Designing adversarial signals against three deep learning and non-deep learning methods

Huang, Yi

2021

Huang, Y. (2021). Designing adversarial signals against three deep learning and non-deep learning methods. Doctoral thesis, Nanyang Technological University, Singapore.
<https://hdl.handle.net/10356/151718>

<https://hdl.handle.net/10356/151718>

<https://doi.org/10.32657/10356/151718>

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Downloaded on 26 Apr 2023 20:17:12 SGT

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

SINGAPORE

DESIGNING ADVERSARIAL SIGNALS AGAINST
THREE DEEP LEARNING AND NON-DEEP
LEARNING METHODS

HUANG YI

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

2021

Designing adversarial signals against three deep learning and non-deep learning methods

Huang Yi

School of Computer Science & Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2021

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

15/11/2020

.....

Date

huang yi

.....

Huang Yi

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

15/11/2020

.....

Date

.....

Prof. Lam Kowk Yan

Authorship Attribution Statement

This thesis contains material from 4 papers published in the following peer-reviewed conferences in which I am listed as an author.

Chapter 3 is published with material from the following papers:

(1)**Huang, Yi**, Adams Wai-Kin Kong, and Kwok-Yan Lam. “Attacking object detectors without changing the target object.” In Pacific Rim International Conference on Artificial Intelligence, pp. 3-15. Springer, Cham, 2019.

AND

(2)**Huang, Yi**, Adams Wai-Kin Kong, and Kwok-Yan Lam. “Adversarial Signboard against Object Detector.” In BMVC, 2019.

The contributions of the co-authors are as follows:

- I came up with the key idea and designed all experiments, implemented the source code and conducted all experiments. I prepared the manuscript drafts.
- Prof Kong discussed with me the key idea and the design details.
- Prof Lam and Prof Kong revised the manuscript.

Chapter 4 is published with material from:

Huang, Yi, Wang, Fan, Adams Wai-Kin Kong, and Kwok-Yan Lam. “New Threats against Object Detector with Non-Local Block”, ECCV, 2020

The contributions of the co-authors are as follows:

- I came up with the key idea and designed experiments, implemented the source code, and conducted all experiments. I prepared the main manuscript drafts.
- Dr. Fan Wang provided the math analysis of effective attack regions and the manuscript of this part.
- Prof Kong discussed with me the key idea.
- Prof Lam and Prof Kong revised the manuscript.

Chapter 5 is published with material from:

Huang, Yi, Adams Wai Kin Kong, and Kwok-Yan Lam. “From the Perspective of CNN to Adversarial Iris Images.” In 2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS), pp. 1-10. IEEE, 2018.

The contributions of the co-authors are as follows:

- I proposed and implemented experiments and prepared the manuscript.

- Kong provided the initial direction, discussed with me the key idea and the design details.
- Prof Lam and Prof Kong revised the manuscript.

15/11/2020

.....

Date

huang yi

.....

Huang Yi

Acknowledgements

I am grateful to all those who have helped and supported me throughout my Ph.D. life.

First of all, I would like to express my heartiest gratitude to my supervisor, Prof. Lam Kwok Yan, for offering me the chance to study here. During the four years of study, he guided me with his immense knowledge and provided me the maximum freedom to dive into the research topic I like. His insightful feedback pushed me to sharpen my thinking and brought my work to a higher level.

Many thanks to my co-supervisor Prof. Adams, Kong Wai Kin, for his precious advice and guidance to my research. His co-supervision is very important for my Ph.D. study. His patience and meticulous suggestions helped me overcome many difficulties and finish this thesis.

I would also like to thank all of the colleagues and friends I met during my four years at NTU. They are a group of people of both friendliness and excellence. Together they formed a vigorous environment, inspiring my spirit every day on the campus.

I would also like to express my appreciation to my boyfriend, Dr. Jun Xie, for his consistent encouragement, love, and always being by my side. Last but not least, I want to thank my parents for their love during the study with the weekly video call. They are always there for me.

Abstract

The widespread adoption of machine learning, especially Deep Neural Networks (DNNs) in daily life, causes great concern about its security property. Szegedy et al.'s study showed that DNNs are vulnerable to adversarial examples, which are images with small, deliberately designed perturbations. The perturbations are called adversarial signals in this thesis. Inspired by their study, researchers are investigating the risks of DNNs models under various adversarial attacks. Though many works have been done on DNN classification models, there are still some models needed to be studied comprehensively. In this thesis, we proposed adversarial attacks against three different types of architectures that have not been thoroughly examined by previous work.

Our first target models are object detectors. Most of the existing works focus on the security property of image classifiers. Only a small number of them concentrated on other widely utilized DNN models. Object detectors, such as Faster R-CNN and YOLO, have numerous applications, including in some critical systems, e.g., self-driving cars and unmanned aerial vehicles. Their vulnerabilities have to be studied thoroughly before deploying them in critical systems to avoid irrecoverable loss caused by intentional attacks. Researchers have proposed some methods to craft adversarial examples for studying security risk in object detectors. All these methods require modifying pixels inside the target objects. These methods are either inapplicable to the physical world or significantly distort the target objects. In this thesis, two new algorithms were proposed to generate adversarial signals that can successfully attack object detectors both digitally and physically without changing the target object itself. The first algorithm generates a type of adversarial signal named adversarial border, which can fool object detectors by being placed

around the border of target objects. This is the first of its kind that does not change pixels within the target object. However, the generated adversarial border did make the target object look weird to humans. Therefore, a new type of adversarial signal, which looks like a signboard, generated by our second algorithm is proposed to make the attack more natural. By being placed below a target object, it can mislead the state-of-the-art object detectors. The experimental results show that they can effectively fool the target object detectors digitally and physically.

Our second target architecture is the non-local block. Current works usually study the risk of the entire model, while the risk of DNNs may be caused by a specific component. The introduction of non-local blocks to the traditional CNN architecture enhances its performance for various computer vision tasks by improving its capabilities of capturing long-range dependencies. However, the usage of non-local blocks may also introduce new threats to computer vision systems. Therefore, it is important to study the threats caused by non-local blocks before directly applying them to commercial systems. In this thesis, two new threats named disappearing attack (DA) and appearing attack (AA) against object detectors with a non-local block are investigated. DA aims at misleading an object detector with a non-local block to make it unable to detect a target object category while AA aims at misleading the object detector into detecting a predefined object category, which is not present in the images. Different from the existing attacks against object detectors, these attacks can be performed in long-range. The experiments show that the proposed attacks can mislead the detector with great probabilities. To explain the threats from non-local blocks, the reception fields of CNN models with and without non-local blocks are studied both empirically and theoretically.

Our third target architecture is IrisCode, a widely used iris recognition algorithm. Most of the existing works focus on studying the adversarial examples against DNNs models. However, there are still some widely applied algorithms not based on DNNs that need to be investigated thoroughly. IrisCode, the most influential iris recognition algorithm, has been extensively employed in national identity programs and border controls. It is vital to fully examine its security issues. How to generate

iris images from its templates is one of such security issues. The existing generation methods synthesized low-quality images with obvious artifacts. Generalizing the definition of adversarial examples in deep learning, we derived a constrained convex minimization formulation to generate adversarial iris images. The experimental results demonstrate that the proposed algorithm can produce iris images, which can be matched with real iris images from a target eye and significantly outperform the previous methods in terms of visual quality.

Contents

Acknowledgements	v
Abstract	vi
List of Figures	xi
List of Tables	xiv
1 Introduction	1
1.1 Motivation	1
1.2 Research Objectives	3
1.3 Major Contributions	4
1.4 Outline of the Thesis	6
2 Literature Review	7
2.1 Adversarial Example against Conventional Machine Learning	8
2.2 Adversarial Example against Image Classification	9
2.2.1 Attack Model	10
2.2.2 White-box Attacks	12
2.2.3 Black-box Attacks	20
2.3 Adversarial Examples against Object Detection	22
2.3.1 Faster R-CNN and YOLO	22
2.3.2 Attacks against Object Detection	23
2.4 Adversarial Examples against Other Systems	26
3 Attacking Object Detectors without Changing the Target Object	28
3.1 Introduction	28
3.2 Adversarial Border	30
3.2.1 Methodology	30
3.2.2 Experimental Results	34
3.3 Adversarial Signboard	41
3.3.1 Methodology	41
3.3.2 Experimental Results	43
3.4 Summary	47

4	New Threats against Object Detector with Non-Local Block	49
4.1	Introduction	49
4.2	Methodology	51
4.2.1	Faster R-CNN and Non-local Block	51
4.2.2	Disappearing Attack	53
4.2.3	Appearing Attack	54
4.3	Experiments	55
4.3.1	Digital Attack	56
4.3.2	Physical Attack	59
4.3.3	Effective Attack Regions	61
4.4	Analysis of the Effective Attack Regions	64
4.5	Extra Experiments on OpenImage Dataset	67
4.5.1	Disappearing Attack	67
4.5.2	Appearing Attack	68
4.6	Summary	68
5	From the Perceptive of CNN to Adversarial Iris Image	73
5.1	Introduction	73
5.2	A Computation Summary of IrisCode and Notations	76
5.3	From a View of Convolutional Neural Network to an Unconstrained Formulation	78
5.4	From Unconstrained Minimization to Constrained Minimization	80
5.5	Experimental Results	82
5.6	Summary	91
6	Conclusion	93
6.1	Major Contributions	93
6.2	Future Work	95
	List of Author's Publications	97
	Bibliography	98

List of Figures

2.1	Poisoning attack and evasion attack	8
2.2	An illustration of adding adversarial perturbation to the original image by FGSM (taken from Goodfellow et al. [1]). The left is an image of a panda from ImageNet. The middle is adversarial perturbations and the right is the adversarial example generated by FGSM.	14
2.3	An illustration of universal adversarial perturbation taken from [2] .	18
2.4	An illustration of NAA adapted from [3]	19
2.5	Adversarial examples generated by different methods for YOLO detector from [4]	24
2.6	An adversarial example taken from DAG [5]	25
2.7	Adversarial examples generated by [6]	25
2.8	Adversarial stop signs produced by robust physical attack [7]	26
3.1	Illustration of Faster R-CNN.	31
3.2	Examples for (A) perturbed training images and (B) perturbed testing images.	32
3.3	The 12 adversarial borders used in the experiments. The adversarial borders in the 1 st -4 th columns are respectively computed from α_1 to α_4 and the adversarial borders in the 1 st -3 rd are respectively obtained at the 100 th , 200 th and 300 th iterations.	35
3.4	Sample frames from the five videos for evaluating digital attack. . .	36
3.5	An adversarial border and the original stop sign on printouts. . . .	39
3.6	Detection results on images with and without adversarial borders. (A) Results from the digital attacks and (B) results from the physical attacks.	40
3.7	Normal signboards collected from the Internet.	43
3.8	The first column is the three adversarial signboards for digital attacks and the second column is their printouts for physical attacks. . . .	44
3.9	Detection results with and without adversarial signboards from Faster R-CNN(ResNet-101)	46
3.10	Sample frames with and without adversarial signboard from the videos in the same group.	47

4.1	Adversarial patches for the (A) disappearing attacks and (B) appearing attacks against the Faster R-CNN-WN. Adversarial patches for the (C) disappearing attacks and (D) appearing attacks against the Faster R-CNN.	55
4.2	Disappearing attack results. (A) The detection results of stop sign from the original Faster R-CNN and (B) the detection results of stop sign from the Faster R-CNN-WN. The first column is the original detection results without the adversarial patches and the second column is the detection results with the adversarial patches.	57
4.3	Appearing attack results, where boat is a target object. (A)-(B) The detection results of boat from the Faster R-CNN-WN. (C) The detection results of boat from the original Faster R-CNN.	60
4.4	The detection results of stop sign from the Faster R-CNN-WN (A) without and (B) with the adversarial patch in disappearing attack. Note that the stop sign closed to the tree is a real stop sign.	61
4.5	The Faster R-CNN-WN detection results of boat in appearing attack.	62
4.6	Two sample images with the grids.	62
4.7	Successful attack rates in different locations, (A) Faster R-CNN-WN and (B) Faster R-CNN. The red boxes indicate the locations of the stop signs. The yellow lines indicate successful attack rates greater than 10% and the green line in (A) indicate successful attack rate greater than 30%.	64
4.8	Disappearing attack results. The first column is the original detection results without the adversarial patches and the second column is the detection results with the adversarial patches.	69
4.9	Appearing attack results. The first column is the original detection results without the adversarial patches and the second column is the detection results with the adversarial patches.	70
5.1	The computational process of IrisCode.	78
5.2	Illustration of the filtering and coding processes in IrisCode as a convolutional neural network.	78
5.3	Images without iris in the UBIRIS.v1 database.	82
5.4	Low quality samples used in the evaluation.	82
5.5	The cumulative distributions of normalized bit error from the (A) UBIRIS.v1 database (B) WVU database.	85
5.6	The ROC curves of the information hiding method and the proposed algorithm for (A) UBIRIS.v1 database (B) WVU database.	86
5.7	Visual comparison of different results. (A) UBIRIS.v1 database and (B) WVU database. There are seven images in each group. The first and the second images are respectively the original normalized iris image and the reference image, X_κ in Eqs. 5.12–5.13. The 3 rd to 7 th images are respectively CAIG, GB, CPC, IH(0.3) and IH(0.7) results.	88

5.8	The distributions of non-reference image quality indexes. The first and the second column are respectively distributions of the UBIRIS.v1 and WVU databases. The 1 st -3 rd rows are respectively distributions of BRISQUE ,NIQE and DIIVINE.	89
-----	---	----

List of Tables

2.1	Summary of adversarial attacks agisnt image classification covered in the section.	13
3.1	The original detection rates of Clip 1 – Clip 5 for digital attacks . .	34
3.2	Detection rates of Faster R-CNN and YOLOv3 for the videos collected in location 1 and location 2 with adversarial borders.	37
3.3	Average successful attack rates of digital attacks	38
3.4	Detection rates of Faster R-CNN and YOLOv3 for the videos collected in location 1 and location 2 with adversarial borders.	39
3.5	The average successful attack rates in digital attacks (%)	45
3.6	The average detection rates in physical attacks (%)	45
4.1	Successful disappearing attack rates (%)	58
4.2	The successful appearing attack rates (%) under different IOD thresholds.	59
4.3	The detection rate of stop sign in disappearing attacks in the physical world.	63
4.4	The appearing rate of boat in appearing attacks in the physical world.	63
4.5	The successful disappearing attack rates (%) for different classes. . .	72
4.6	The successful disappearing attack rates (%) for different classes. . .	72
5.1	Comparisons of different methods using reference quality image indexes. Each value in the table is the mean of an index for one database.	90
5.2	Comparisons of different methods using non-reference quality image indexes. Each value in the table is the mean of an index for one database.	91

Chapter 1

Introduction

1.1 Motivation

With the dramatic improvement in hardware and exponential growth of available data in the last decade, machine learning, in particular, deep learning has shown promising results in many areas, such as image classification [8, 9], speech recognition [10, 11], and language translation [12]. Consequently, a large number of machine learning algorithms have been deployed in various systems. For example, Amazon applies machine learning in their platform to offer personal recommendations [13]. Google search uses machine learning to improve users' searching experience [14]. Online payment companies such as PayPal use machine learning methods to fight fraud [15]. Apart from these applications in technology companies, machine learning has also been widely adopted in financial market analysis, biometric recognition, malware classification, drug discovery, etc.

The widespread use of machine learning technologies improves the quality of our life. However, they have some severe security risks. In general, a machine learning algorithm is designed to learn a model from data (training data) under an unknown probability distribution for a particular task, e.g., classification, and the learned

model is applied to unseen data (testing data). The common hypothesis in machine learning algorithms is that the training data and the testing data are under the same distribution. This hypothesis can be easily broken by an attacker through manipulating the input data according to specific vulnerabilities of the machine learning algorithms. Due to the extensive applications, attacking machine learning algorithms becomes pretty lucrative. For example, by attacking them, swindlers can escape from fraud detection; forgers can crack biometric systems and hackers can distribute malware across the Internet. However, lots of machine learning algorithms were designed without consideration of adversaries. Therefore, it is vital to study the security properties of machine learning algorithms and design secure algorithms in an adversarial environment.

A recent publication [16] showed that Deep Neural Networks(DNNs) are vulnerable to small, deliberately designed perturbations, which are called adversarial signals in this thesis. Input images with adversarial signals are named adversarial examples. Since adversarial examples are the final input of the target attack model, designing adversarial signals is referred to as generating adversarial examples in this thesis. An adversarial example looks the same as the original image, but the image classifier will output a wrong label for it. What's worse is that the attacks based on adversarial examples are transferable between different classifiers [16]. More clearly, adversarial examples designed to fool a target DNN model can also fool other DNN models. It means that an attacker does not need precise information about the algorithm they want to attack. They can train a substitute DNN model and generate adversarial examples against the substitute model. The generated adversarial examples have a high probability of misleading a target DNN model. Inspired by Szegedy et al. [16], researchers proposed different methods for generating adversarial examples against DNNs. Their works identify potential risks of DNNs, reduce the number of possible zero-day attacks, and provide vital

information to develop countermeasures. Though numerous attacks have been proposed, the security risks of some architectures have not been examined thoroughly. Therefore, we aimed at proposing adversarial attacks against architectures that have not been well studied before. In particular, we explored three different types of architectures: object detection, non-local blocks, and iris recognition.

1.2 Research Objectives

The purpose of this thesis is to explore new threats to architectures that have not been thoroughly studied. Three representative architectures are selected under different perspectives as follows:

- Most of the existing works concentrated on investigating the risk of image classification. Only a small number of them paid attention to other widely applied DNN models. Object detectors have various applications such as face recognition, people counting, self-driving cars. It would be risky to directly use them in a fully automatic environment before their security property is thoroughly studied. There are a few works about attacking object detection systems. These attacks either cannot generalize to the physical world or significantly change target objects, which makes the adversarial examples hard to recognize even by human beings. Therefore, in this thesis, we investigated two different adversarial attacks against object detectors which can be performed both digitally and physically without changing the target object.
- Previous works usually studied the risk of the entire model, while a particular risk of DNNs may be caused by a specific component. The introduction of non-local blocks to the traditional CNN architecture enhances its performance for various computer vision tasks by improving its capabilities of capturing long-range dependencies. However, the usage of non-local blocks

may also introduce new threats to computer vision systems. Therefore, it is important to study the threats caused by non-local blocks before directly applying them to commercial systems. In this thesis, two new threats named disappearing attack (DA) and appearing attack (AA) against object detectors with a non-local block are investigated. Different from the existing attacks against object detectors, these threats can be performed in long-range cases.

- Recent works focus on studying the adversarial examples against DNNs models. However, some widely applied models are not based on DNNs. The security risks of these conventional models under adversarial example attacks have not been explored. Up until now, the iris recognition systems, which have been widely used for national identity and border control, are still based on IrisCode (an iris recognition method, whose template is also called IrisCode in the biometric community). Hence we selected IrisCode as the target algorithm and tried to generate iris images with target IrisCode. The existing generation methods can produce images with target IrisCode and the synthetic images can be used to match other real iris images. However, their quality is low with obvious artifacts. Different from them, we aimed at generating high-quality iris images. More concretely, given a target IrisCode and an original iris image with a different IrisCode, our goal is to generate an iris image with the target IrisCode, and the generated image should look the same as the given image by human eyes.

1.3 Major Contributions

Our main contributions can be stated as follows:

- **Adversarial border:** We propose an adversarial border, which is a new type of adversarial signal against object detectors. To the best of our knowledge,

this is the first of its kind in attacking object detectors without changing the target object. The experimental results show that the adversarial border can effectively fool Faster R-CNN and YOLOv3 digitally and physically. They also demonstrate that an adversarial border can be used to perform white-box and black-box attacks.

- **Adversarial signboard:** Though an adversarial border can successfully fool object detectors, it makes the target object looks unnatural to human. We propose an adversarial signboard to attack object detectors without changing the target object and to overcome the above-mentioned problem. It is placed below the target object with a distance and looks more natural than an adversarial border. Experiments on five object detectors have been carried out and demonstrated that adversarial signboard can perform digital and physical attacks in both white-box and black-box settings.
- **New Threats against Object Detector with Non-local Block:** We study the threats caused by non-local blocks in object detectors. It is the first study analyzing risks from non-local blocks. We propose two different attacks against object detectors with a non-local block: disappearing attack (DA) and appearing attack (AA). The goal of DA is to make the target model unable to detect a target object category while the goal of AA is to mislead the target model into detecting a predefined object category, which is not present in images. Different from the existing attacks against object detectors, these threats can be performed in long-range cases. This means that the target object and the universal adversarial patches learned from the proposed algorithms can have a long-distance between them. The experiments show that the universal patches can mislead the detector with great probabilities.
- **Adversarial Iris image:** we propose a constrained convex minimization formulation with an inputted iris image selected by a search function as an information carrier to generate iris images from its templates. The experimental

results demonstrate that the proposed algorithm can generate high-quality iris images and significantly outperform the previous methods in terms of visual quality and six image quality metrics. They can also match with other real iris images from the target persons.

1.4 Outline of the Thesis

The rest of the thesis is organized as follows: Chapter 2 reviews the related work about adversarial examples. Chapter 3 introduces two new types of attack against object detectors. Chapter 4 investigates the threats caused by non-local blocks in object detectors. Chapter 5 introduces a constrained convex minimization formulation to generate iris images from its templates. Chapter 6 gives conclusive remarks and discussions.

Chapter 2

Literature Review

Though the phenomenon that DNNs can be easily fooled by adversarial examples was first found by Szeged et al. [16] in 2014, it was not the start point of adversarial examples. Adversarial attacks against machine learning have been discussed for more than ten years. These attacks can be divided into different types according to an adversary's goal, knowledge, capabilities, and attack strategies. Among all the proposed scenarios, poisoning attacks and evasion attacks are two of the most frequently studied scenarios. In a poisoning attack, adversaries can contaminate the training data by injecting carefully designed samples, which will compromise the whole learning phase. This scenario is suitable for some adaptive or online learning algorithms, which will be retrained on data collected during test time. In an evasion attack, attackers have no access to the training data, meaning that they cannot influence the training stage. Usually, they attack the system by crafting testing data. These carefully designed testing data are named adversarial examples later in deep learning. Figure 2.1 is an illustration of these two types of attacks. Compared to poisoning attacks, evasion attacks are more common and realistic. In deep learning, most of the models are trained offline. It's not easy to contaminate the training data. Hence we only concentrate on attacking at test time in this thesis.

FIGURE 2.1: Poisoning attack and evasion attack

2.1 Adversarial Example against Conventional Machine Learning

In 2004, Dalvi et al. [17] and Lowd et al. [18, 19] showed that the spam filtering system based on linear classifiers could be easily fooled by carefully crafted spam emails without depriving the readability of the spam message. These are the earliest adversarial examples against spam filtering systems. Nelson et al. [20] demonstrated that by systematically querying a spam filter, the adversary can construct undetected spam emails. Biggio et al. [21] proposed a gradient-based algorithm for the evasion of classifiers with differentiable discriminant functions such as SVM and neural networks. They simulated attack scenarios at different risk levels based on the attacker's knowledge of the classifiers and their capability. They studied the effectiveness of the devised attacks on a handwritten digit classifier and PDF file malware detector under the above-mentioned scenarios. Srndic and Laskov [22] investigated the effectiveness of the evasion attacks on a deployed system PDFRATE under the assumption that the attacker has no specific insider information. The experimental result showed that the classification scores drop significantly even when it is exposed to some simple attacks. Venugopalan and Savvides borrowed the idea from information hiding to embed iris information from a given iris template into a real iris image [23]. Their experimental results showed that the generated images

can match with iris images collected from the eye, where the template is extracted from. Evasion attacks in conventional machine learning concentrated on security tasks such as spam detection, malware detection, or biometric authentication. In these attacks, the attackers' goal is to make malicious input being classified as benign while maintaining their malicious functions. For example, the attacker in [17] crafted the spam message to pass the spam filter without removing their ability in spreading the spam information. Therefore, in these attacks, the requirement for adversarial perturbations is 1) to mislead the classifier and 2) to keep the original information. Most recent works about adversarial examples focus on computer vision tasks. Based on the requirements, the adversarial image and original input without perturbation should be very similar. In other words, they are not distinguishable by human eyes. Therefore, adversarial perturbations are required to be minimal or sufficiently small.

2.2 Adversarial Example against Image Classification

With a large amount of online data and the rapid growth of computational power, deep learning has achieved great performance in various domains, such as computer vision, speech recognition, and natural language processing. With deep learning, an increasing number of new applications of computer vision have been introduced in the real world and becoming parts of our daily lives, for example, the face recognition systems deployed in ATMs in China [24] and self-driving car [25]. The broad adoption of deep learning in computer vision tasks, especially in some life crucial areas, causes great concern about its security property. A great deal of work has been done to study the security of deep learning. It can be summarized into three research directions: (1) designing new attack strategies and countermeasures against adversarial examples, (2) estimating boundaries of DNNs against

zero-day (new) attacks, and (3) explaining the existence of adversarial examples and their transferability. From the application perspective, designing adversarial examples is arguably the most important direction for the reason that it reduces the number of possible zero-day attacks and provides critical information to develop countermeasures. In 2014, Szegedy et al. [16] first showed that the state-of-the-art image classifier could be fooled by deliberately designed adversarial perturbations. Following Szegedy et al.'s research, researchers proposed numerous attacks against various tasks. Since most of the proposed attacks focused on image classification and the attacks against other applications are extended from the attacks against image classification, we first review representative approaches for generating adversarial examples against image classification and then introduce adversarial attacks against object detection and other applications in this section.

2.2.1 Attack Model

Biggio et al. [26] provided a general attack model based on the framework proposed by [27]. The model takes different kinds of attacks (attacks both at training time and testing time) into consideration and showed a standard procedure to systematically analyze attacks against machine learning algorithms. The model consists of four different aspects: attackers' goal, knowledge of the target model, the capability of manipulating the input data, and their attack strategy. Attack using adversarial examples is only one kind of adversarial attacks against machine learning algorithms. To introduce these attacks systematically, we define a threat model for them in this section. The model is based on [26] and only covers adversarial examples for image classification.

Attacker's Goal: the goal of misleading the classifier can be further divided into two types: untargeted and targeted.

- Untargeted, where the goal of the attack is to fool the classifier into predicting any incorrect class label. It can be defined as:

$$\begin{aligned} \min \quad & D(x, x') \\ \text{s.t.} \quad & y(x') \neq l, x' \in [0, 1]^n \end{aligned} \tag{2.1}$$

where $D(\cdot)$ is a distance metric, e.g. L_2 norm. x is a clean input with a true label l and x' is an adversarial exmple. $y(x)$ is the output label of a classifier.

- Targeted, where the goal of the attack is to make the classifier output a specific class. It is written as:

$$\begin{aligned} \min \quad & D(x, x') \\ \text{s.t.} \quad & y(x') = t, x' \in [0, 1]^n \end{aligned} \tag{2.2}$$

where t is the target class label.

Attacker' knowledge: the attackers' knowledge can be classified into:

- White-box, where the attacker has full knowledge of the model, including the archietcture, the model parameters and the training data. This is the easiest scenario for an attacker. It means that an attacker can completely replicate the model under attack.
- Black-box, where the attacker has no knowledge of the model and can only get the output of the target model. This is the most difficult scenario for an attacker. However, due to the transferability of adversarial examples in DNNs, black-box attacks become much easier [28]. The attackers just need to train a substitute model to generate adversarial examples. According to the transferability, the adversarial examples have a high probability of attacking the target model.

- Gray-box, where the attacker may only have partial knowledge of the model. As gray-box attacks are uncommon in the literature, we do not include it in the taxonomy.

Attack strategy: the attack strategy can be categorized into:

- Optimization based, where the attacker defines the attack as an optimization problem and solves it by optimization algorithms.
- Generative based, where the attacker trains a generative model to approximate the probability of adversarial perturbations and to create new adversarial examples.

Table 2.1 summarizes attacks against image classification covered in this section based on the attack model outlined above. Our review of adversarial attacks is based on the adversary’s knowledge.

2.2.2 White-box Attacks

According to our statistics, most of the attacks are under the white-box assumption. Following the attack model, these attacks are divided into two categories: optimization based attacks and generative based attacks.

Optimization based attacks

L-BFGS Attack: Szgedy et al. [16] showed that images with adversarial perturbations can successfully mislead the state-of-the-art deep neural networks. They found an approximation of the minimizer in equation 2.2 using a box-constrained Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS). More clearly, they

TABLE 2.1: Summary of adversarial attacks agisnt image classification covered in the section.

Attack	Attackers' goal	Attackers' knowledge	Attack strategy
L-BFGS [16]	T	W	Optimization based
FGSM [1]	U	W	Optimization based
FGM [29]	T	W	Optimization based
R-FGM [30]	T,U	W	Optimization based
BIM [31]	U	W	Optimization based
ILLCM [31]	T	W	Optimization based
MI-FGSM [32]	U	W	Optimization based
JSMA [33]	T,U	W	Optimization based
DeepFool [34]	U	W	Optimization based
C&W Attack [35]	T,U	W	Optimization based
UAP [2]	U	W	Optimization based
ATN [36]	T,U	W	Generative based
NAE [3]	U	W,B	Generative based
GAP [37]	T,U	W	Generative based
UAE [38]	T	W	Generative based
SBA [28]	T	B	Optimization based
EBA [39]	T,U	B	Optimization based
ZOO [40]	T,U	B	Optimization based
SPSA [41]	T,U	B	Optimization based

W: White-box Attack

B: Black-box Attack

T: Targeted

U : Untargeted

found the approximation of $D(x, x')$ by solving the following optimization problem:

$$\begin{aligned}
 \min_{\eta} \quad & c \|\eta\| + J(x', l') \\
 s.t. \quad & x' \in [0, 1]
 \end{aligned} \tag{2.3}$$

where $\eta = x - x'$ and $J(\cdot')$ is the loss function of the image classifier. c is a constant. They performed a line search to find the minimum $c > 0$ for which the minimizer η of equation (2.3) satisfies $y(x + \eta) = t$.

FGSM Attack: The line search in L-BFGS Attack is time-consuming. Goodfellow et al. [1] proposed an attack named Fast Gradient Sign Method (FGSM) to quickly

FIGURE 2.2: An illustration of adding adversarial perturbation to the original image by FGSM (taken from Goodfellow et al. [1]). The left is an image of a panda from ImageNet. The middle is adversarial perturbations and the right is the adversarial example generated by FGSM.

find a suitable adversarial perturbations direction. The adversarial examples generated by FGSM can be expressed as:

$$x' = x + \epsilon \cdot \text{sign}(\nabla_x J(x, l)) \quad (2.4)$$

where $\nabla_x J(x, l)$ is the first derivative of the loss function with respect to the input x . ϵ is a parameter controlling the magnitude of the perturbation. Figure 2.2 is an illustration of adding adversarial perturbation to an image to fool GoogLeNet [42] by FGSM.

The variant of FGMS: Rozsa et al. [29] proposed a variant of FGSM named Fast Gradient method (FGM), where they replaced the sign of gradient with a raw gradient to allow larger local difference. To avoid the defense strategy of gradient masking, Tramèr et al. [30] added small random perturbations to the input before applying the adversarial perturbations generated by FGSM (R-FGSM). Experimental results show that R-FGSM is stronger than FGSM when attacking adversarially trained Inception ResNet v2 and Inception v3 models.

BIM and ILLCM: Kurakin et al. [31] extended FGSM to real-world applications by running a finer optimization with multiple iterations. They named the method as Basic Iterative Method (BIM). The process to generate adversarial examples in

BIM can be expressed as follow:

$$\begin{aligned} x'_0 &= x \\ x'_{i+1} &= \text{Clip}_{x,\epsilon} \{x'_i + \alpha \cdot \text{sign}(\nabla_x J(x'_i, l))\}, i = 0, 1, 2, \dots, n \end{aligned} \quad (2.5)$$

where n is the number of iterations and α is the step size of each iteration. $\text{Clip}_{x,\epsilon} \{x'\}$ constrains the value of x' in each pixel to be in ϵ neighborhood of the input image x . The clipping process at the coordinate (u, v, w) can be written as:

$$\text{Clip}_{x'_i,\epsilon} \{x'_{i,(u,v,w)}\} = \min \{255, x_{(u,v,w)} + \epsilon, \max \{0, x_{(u,v,w)} - \epsilon, x'_{i,(u,v,w)}\}\} \quad (2.6)$$

To attack a specific class, Kurakin et al. proposed a variant of BIM named Iterative Least-likely Class Method (ILLCM). The goal of ILLCM is to mislead the classifier into predicting an adversarial example as the target label. They chose the least likelihood of the prediction as the target class to prevent the classifier from finding trivial adversarial examples for which the true classes are similar.

MI-FGSM: Dong et al. [32] proposed to boost the performance of BIM and ILLCM by exploiting momentum during the optimization process. They claimed that the momentum helps stabilize the update direction and helps avoid poor local minimum when applied to optimization based attacks. MI-FGSM won the NIPS 2017 Competition on Adversarial Attacks and Defenses [43] under targeted and untargeted attack categories. The experimental result also showed that adversarial examples generated by MI-FGSM transfer better than those generated by FGSM and BIM on various ImageNet model.

JSMA: Papernot et al. [33] designed an attack named Jacobian-based Saliency Map Attack (JSMA) which considers the difference of the influence from different pixels. JSMA uses a saliency map to rate each pair of input pixels by their influence on the classifier's prediction and modifies the pair of input pixels which makes the most

significant change to the output. The crafting algorithm is described in Algorithm 1. More details about calculating the saliency map and crafting the pixels are provided in the original paper [33]. Compared to FGSM, JSMA just perturbed a small part of the input and achieved a higher adversarial success rate. However, JSMA is much slower than FGSM due to the high computation cost.

Algorithm 1: Jacobian-based Saliency Map Attack

Input: x : clean image, t : target class, f : a classifier mapping input image to a discrete probability set, y : a classifier mapping input image to a label, $Iter_{max}$: maximum number of iterations, θ : change made to pixels, $saliency_map(\cdot)$: the operation generating a map defining which input features will be modified at each iteration.

Output: Adversarial example x'

```

1  $x' \leftarrow x, i \leftarrow 0$  ;
2 Search domain is all pixels  $\Gamma = (1, 2, \dots, |x|)$  ;
3 while  $y(x') \neq t$  and  $i < Iter_{max}$  and  $|\Gamma| \neq 0$  do
4   Calculate  $\nabla_{x'} f(x')$  ;
5    $p_1, q_1 = saliency\_map(\nabla f(x'), \Gamma, t)$  ;
6   Modify  $p_1$  and  $q_1$  in  $x'$  by  $\theta$ 
7   if  $p_1 == 0$  or  $p_1 == 1$  then
8     Remove  $p_1$  from  $\Gamma$ 
9   if  $q_1 == 0$  or  $q_1 == 1$  then
10    Remove  $q_1$  from  $\Gamma$ 
11    $i \leftarrow i + 1$ 
12 final ;
13 return  $x'$ ;

```

DeepFool: For a linear binary classifier, the distance of an input to the decision boundary can be analytically computed. Moosavi-Dezfooli et al. [34] generalized it to DNNs and proposed Deepfool to find the shortest distance between the original input and the decision boundary of the classifier. This attack generates adversarial examples by the following steps: (1) linearizing per-class decision boundaries around the current point, (2) identifying the class whose linearized decision boundary is closet to the current point, (3) moving the current point to the boundary point found in step(2), (4) repeating step (1)-(3) until getting a point being misclassified by the target classifier. DeepFool generates much smaller perturbations

than FGSM and it is much faster than L-BFGS.

C&W Attack: Carlini and Wanger [35] constructed a stronger attack (C&W's attack), which is effective against various defense mechanisms. They expressed equation 2.2 in another form:

$$\begin{aligned} \min_{\eta} \quad & \|\eta\| + c \cdot g(x + \eta) \\ \text{s.t.} \quad & x + \eta \in [0, 1]^n \end{aligned} \tag{2.7}$$

where function g satisfies the following requirement: $g(x') \leq 0 \leftrightarrow y(x') = t$. The attack introduced a new variable w where $\eta = \frac{1}{2}(\tanh(w) + 1) - x$. Instead of directly optimize η , they optimized w to avoid the box constrain. General optimization algorithms such as SGD and Adam can solve this problem. They showed that their attack outperformed FGSM, BIM, JSMA, and DeepFool both in average distortion and attack success rate. C&W attack can also transfer to models trained with defensive distillation [44], which is one of the popular defense methods.

UAP: Universal Adversarial Perturbation (UAP) [2] aimed to generate universal perturbations, which can be added to various images to cause misclassification. Figure 2.3 is an illustration of UAP. The universal perturbations are generated by accumulating perturbations calculated iteratively over a set of images. In each iteration, they use DeepFool to find the minimal perturbation $\Delta\eta$ for an input sample $x + \eta$. Then η is updated by $\eta = \eta + \Delta\eta$ and taken as the new perturbations for another data point in the next iteration. This loop stops when most of the data samples are fooled. The UAP is shown to be existed in various models including CaffeNet [45], GoogLeNet [42], ResNet[46] and VGG[47]. The authors also showed that UAP is highly transferable between models.

Generative based attacks

FIGURE 2.3: An illustration of universal adversarial perturbation taken from [2]

Generative based attacks learn a model to approximate the distribution of adversarial perturbations and sample new perturbations from generative models. Usually, the adversarial examples are strong in terms of attack. However, it costs much more resources to perform generative based attacks than optimization based attacks.

ATN: Adversarial Transformation Networks (ATN) [36] either output a specific adversarial perturbation or directly generate an adversarial example for an input image. The transferability of adversarial examples generated by ATN is quite limited. Baluja and Fischer [36] enhanced the transferability by fooling multiple models at the same time. ATN is easy to train and can generate diverse adversarial examples.

NAA: Zhao et al. [3] exploited Generative Adversarial networks to generate more natural adversarial examples (NAE). They trained a WGAN [48], in which the generator G maps a latent representation z to an instance of natural input x . They also learned an inverter network I which maps instance x back to latent

FIGURE 2.4: An illustration of NAA adapted from [3]

representation z . After having a trained G and I , a natural adversarial example $x' = G(z')$ can be generated by searching for a misclassified latent instance z' :

$$\underset{z'}{\operatorname{argmin}} \|z' - I(x)\| \quad s.t. \quad f(G(z')) \neq f(x) \quad (2.8)$$

where f is the target model. Zhao et al. proposed two methods for finding x' : iterative stochastic search and hybrid shrinking search [3]. They applied NAA to different domains including image classification, text entailment, and machine translation. NAA can also be applied to a black-box attack as it does not require gradient information.

GAP: Poursaeed et al. [37] proposed a generative model that can produce universal and image-dependent adversarial perturbations (GAP) for both targeted and untargeted attacks. They showed that GAP can attack both classification and segmentation models using similar architectures. The perturbations generated by GAP achieved high fooling rates with small perturbation norms.

UAE: Song et al. [38] proposed a new type of adversarial example named unrestricted adversarial examples (UAE) that are not restricted to small norm bounded perturbations. UAE directly synthesizes adversarial examples from scratch using conditional generative modes. To model the class-conditional distribution over input samples, they trained an Auxiliary Classifier Generative Adversarial Network (AC-GAN). Then they found adversarial images that are under the generative

models and are misclassified by a target model by searching the AC-GAN latent space conditioned on a specific class. UAE is effective even on adversarial training and certified defense methods designed for traditional adversarial attacks.

2.2.3 Black-box Attacks

There are two different strategies to carry out a black-box attack. The first strategy is to train a surrogate model and design adversarial examples for it. According to the transferability, the designed adversarial examples have a high probability of fooling a target model. The second strategy is to estimate the gradient based on the output. In the second strategy, an attacker only has access to the output and can have no prior knowledge of the training data.

Substitute Black-box Attack (SBA): Papernot et al. [28] proposed one of the early black-box attacks leveraging the transferability of adversarial examples. They trained a surrogate model with a synthetic dataset in which the inputs are generated by the adversary, and the outputs are labels assigned by the target DNN. Then they used FGSM and JSMA to generate adversarial examples. The success of this approach depends on the similarity between the surrogate model and the target DNN. Papernot et al. introduced a heuristic to generate synthetic training inputs, which ensures that substitute DNN is an approximation of the targeted DNN. They performed SBA on real-world image recognition systems from Amazon, Google, and MetaMind on MINIS, and achieved high success attack rates.

Ensemble-based Black-box Attack (EBA): Liu et al. [39] conducted an extensive study of the transferability over large models and large datasets. They noticed that non-targeted examples are much easier to transfer than targeted examples. To improve the transferability of targeted adversarial examples, they proposed ensemble-based approaches. Using such approaches, they generated non-targeted

and targeted adversarial examples that can successfully attack a black-box image classification system named Clarifai.com.

Zeroth Order Attack (ZOO): Instead of training a surrogate model, Chen et al. [40] proposed a black-box attack named Zeroth Order Optimization(ZOO) which generates adversarial examples by only querying a target model. They proposed loss functions J for both target and untargeted attacks modified from [35]. Then they approximated the gradient and Hessian of the defined loss function with respect to the input using Finite Difference numerical estimates as below:

$$\frac{\partial L}{\partial x} \approx \frac{J(x + h \cdot e_i) - J(x - h \cdot e_i)}{2h} \quad (2.9)$$

$$\frac{\partial^2 L}{\partial x^2} \approx \frac{J(x + h \cdot e_i) - 2J(x) + J(x - h \cdot e_i)}{h^2} \quad (2.10)$$

where h is a small constant and e_i is a standard bias vector with only the i^{th} element being one. ZOO is able to generate adversarial examples without access to the target DNN. Chen et al. showed that ZOO is effective on models trained by defensive distillation and achieves comparable performance as C&W Attack.

SPSA: Uesato et al. [41] use SPSA [49] which is a numerical gradient estimator more efficient than the Finite Difference Method to estimate gradients of a target model similar to ZOO. The experimental result showed that SPAS is stronger than ZOO and is effective against defense strategies proposed in [50–52].

2.3 Adversarial Examples against Object Detection

2.3.1 Faster R-CNN and YOLO

YOLO and Faster RCNN are two state-of-the-art object detection algorithms. To reveal their potential risks and improve their robustness, researchers tried to generate adversarial examples for misleading them. For understanding these attack methods, we briefly introduced YOLO and Faster RCNN first.

YOLO: Redmon et al. [53] designed a simple and fast model named YOLO, which allows real-time object detection. The model divides the input image into an $S \times S$ grid. Each cell in the grid predicts B bounding boxes with a confidence score for each prediction. Each bounding box is represented by its center coordinate, width, and height. In training, the confidence score is the product of the probability of an object in the cell and the Intersection Over Union (IOU) between the predicted and the ground truth boxes. In each cell, YOLO also estimates probabilities of C predefined classes. The final layer outputs a tensor v of size $S \times S(C + B \times 5)$. Non-Maximum Suppression (NMS) is applied at the end of the network to merge highly-overlapping bounding boxes with the same class label into a single one.

Faster R-CNN: Ren et al. [54] introduced a state-of-the-art object detection algorithm named Faster Region-based Convolutional Network (Faster R-CNN). It adopts a two-stage detection strategy. In the first stage, a CNN model takes the entire image as input and produces feature maps. Then a region proposal network (RPN) is used to generate bounding boxes called region proposals that may contain objects. The region proposals and feature maps are taken as input in the second stage. A 3-by-3 window slides all the feature maps and outputs a feature vector linked to two fully-connected layers, one for box-regression and one for box-classification. Finally, the network outputs the classification results and refined

bounding box coordinates for each region proposal. NMS method is applied at the end of the network to remove duplicated predictions.

2.3.2 Attacks against Object Detection

In classification, the classifier outputs a vector, of which each element represents the probability of the input image belonging to a specific class. In training, there is a one-hot vector, whose length is equal to the number of classes and all elements are zeros, except the one corresponding to the ground true class with the value of one. Adversarial examples are generated by maximizing the difference between the output vector and the hot vector for an untargeted attack. In object detection, there is no such one-hot vector and the network output contains not only class predictions but also position information of multiple objects. Hence the attack strategies for image classification cannot be directly applied to fool detectors.

Lu et al. [4] demonstrated attacks against YOLO using L-BFGS, FGSM, and BIM. The outputs vector v in YOLO stores the positions and class probabilities of all detected objects. When all elements in the output vector approach zero, it means that nothing is detected. Lu et al. set a vector y_t which has the same length as the output vector v and each element has the same value of $1/\text{length}(v)$. Since the size of v is relatively large, each element in y_t is almost zero. Using y_t as a target, they employed L-BFGS, FGSM, and IBM to generate adversarial examples and carried out physical attacks by printing out these adversarial images. The experimental results showed that their method can successfully mislead YOLO digitally, but were ineffective in the physical world. Figure 2.5 is an original image and adversarial examples generated by L-BFGS, FGSM, and IBM.

Xie et al. [5] designed an attack named Dense Adversary Generation (DAG) for semantic segmentation and object detection. Their target object detector was Faster R-CNN [54]. Let I be an image containing N positive proposals generated

FIGURE 2.5: Adversarial examples generated by different methods for YOLO detector from [4]

by region proposal networks. Their goal is to confuse the detector such that all predictions for these positive proposals are wrong. Consequently, they specified an adversarial label randomly sampled from other incorrect classes for each target proposal, suppressed the confidence of the original correct class, and increased the confidence of the desired incorrect class for each target. Figure 2.6 is an adversarial example generated by DAG.

Lu et al. [6] presented an algorithm that just needs to modify the target object instead of the whole image to fool Faster R-CNN. A stop sign was used as a target object to examine the algorithm. To make the adversarial pattern robust, they selected a set of diverse frames F which contain a stop sign from a video to train their algorithm. The stop sign in one of the frames was represented as a texture map in a root coordinate system. Using the eight vertices on the stop signs as correspondences, a viewing map that maps the texture in the root coordinate system to the stop sign in the training frame was estimated. To produce an adversarial example, they minimized the mean score for the stop sign produced by Faster RCNN in all image frames which were obtained by superimposing the texture on all the frame F using the viewing map. Figure 2.7 is three adversarial stop signs generated by their method. All three patterns reliably fool detectors when mapped into videos. However, physical instances of these patterns are not equally successful. The first two stop signs on physical objects only occasionally fool Faster RCNN. The third

FIGURE 2.6: An adversarial example taken from DAG [5]

FIGURE 2.7: Adversarial examples generated by [6]

one, which has a much more extreme pattern, is more effective. However, it is hard for a human to recognize it as a stop sign.

The Robust Physical attack [7] is inspired by the change-of-variable attack [87] and the Expectation over Transformation (EOT) technique [132]. The idea of EOT is to add random distortions in each training iteration to make the adversarial perturbation more robust. They devised a variant of change-of-variable attack which

FIGURE 2.8: Adversarial stop signs produced by robust physical attack [7]

is more robust to distortions by incorporating EOT into the original method. Then they applied it to the classification part of Faster RCNN. As described in Section 2.3.1, Faster R-CNN adopts a 2-stage approach. The region proposal network in the first stage outputs many region proposals and the network in the second stage classifies each of the region proposals. They set a target example for each of the region proposal and then simultaneously attacked all the classifications of each region proposal by the devised attack method. They performed their attack on the stop sign and only allowed the perturbation to change the red part of the stop sign, leaving the white text intact which helps humans recognize it easily. Figure 2.8 shows the three adversarial stop signs generated by them. Figure 2.8 (A) and Figure 2.8 (B) were used for targeted attacks. The target class were person and sports ball respectively. Figure 2.8 (C) was used for an untargeted attack.

The above-mentioned attacks against object detectors are ineffective in the physical attack or require very significant modification of target objects [6, 7], which makes them hard to be recognized by human eyes.

2.4 Adversarial Examples against Other Systems

There are also adversarial examples against some other systems. Tabacof et al. [55] and Kos et al. [56] proposed attacks against generative image models which can

mislead them to generate target images completely different from the expected images. Papernot et al. [57] demonstrated that the algorithms generating adversarial examples for feed-forward neural networks can be adapted to fool Recurrent Neural Networks (RNN). Huang et al. [58] and Lin et al. [59] applied FSGM to attack the reinforcement learning model. Some researchers tried to generate adversarial examples for physical attacks against real-world applications. Sharif et al. [60] defined a novel class of attack that is physically realizable and inconspicuous. They provided a method to design adversarial eyeglass frames to attack the state-of-the-art face recognition system [61]. Attackers wearing the adversarial eyeglasses can successfully cause misclassification of the face recognition systems. Xu et al. [62] launched an attack to evade two PDF malware classifiers, PDFrate, and Hidost which are widely used in the real world for malware detection. Although there are some works on adversarial examples for other tasks, the main focus is image classification. More studies need to be done for other tasks, especially for the tasks which are widely adopted in the real world.

Chapter 3

Attacking Object Detectors without Changing the Target Object

3.1 Introduction

With the rapid development of deep learning and computational hardware, as well as the availability of big image and video datasets, the performance of computer vision methods has been significantly improved in recent years. Many of them have been deployed to commercial and military applications. Nevertheless, as with other cyber-physical systems, computer vision systems are also subject to attacks. Szegedy et al. [63] found that even though deep neural networks (DNN) based image classifiers achieve great performance, compared with the traditional hand-crafted feature methods, they are vulnerable to adversarial examples, which are almost the same as their original images to the naked eyes. Their study draws great concerns from both computer science researchers and engineers, in particular those working in the privacy and security-critical industries. Papernot et al. [64]

further discovered that adversarial examples are transferable, meaning that an adversarial example derived from one network can fool other networks trained on different datasets and even with different architectures. Their findings indicated that attackers can use adversarial examples to perform black-box attacks, even when they do not have knowledge about the target network.

In order to identify potential risks and to eventually improve the safety of DNN-based systems, researchers put great efforts on investigating adversarial examples. It can be summarized into three research directions: (1) designing new attack strategies and countermeasures against adversarial examples, (2) estimating boundaries of DNNs against zero-day (new) attacks, and (3) explaining the existence of adversarial examples and their transferability. From the application perspective, designing adversarial examples is arguably the most important because it reduces the number of possible zero-day attacks and provides critical information to develop countermeasures. Following Szgedy et al.'s research in 2014, researchers proposed different methods to generate adversarial examples. Most of them focused on DNN image classifiers and succeeded in misleading the classifiers with high misclassification rates digitally or physically. A question then arises: is it possible to attack more complex computer vision systems, such as those for object detection digitally and physically? Object detection has various applications, for instance, face detection, people counting, self-driving cars, etc. If an object detector is vulnerable to adversarial examples, it may not be wise to use it in a fully automatic environment before corresponding countermeasures are developed. Therefore, it is important and necessary to study the impact of adversarial examples on object detectors.

Object detection is different from image classification. Image classifiers output a class label for an input image, while object detectors localize multiple objects in an image and predict their class labels. Generally, object detectors produce multiple bounding boxes internally for each object and use non-maximum suppression

or other techniques to select one bounding box for each object. Attacking object detectors is more difficult than attacking image classifiers because adversarial examples have to cause significant errors on all the internal bounding boxes [4]. Some researchers successfully attacked object detectors by changing every pixel in entire images, which generally cannot be implemented as physical attacks. Some researchers changed target objects extensively to achieve effective digital and physical attacks. Their adversarial examples are hard to be classified even for human beings. Is it possible to design adversarial examples, which are effective for digital and physical attacks, against object detectors without modifying target objects? In this chapter¹, two algorithms are designed to craft adversarial examples named adversarial border and adversarial signboard. These two types of attacks can attack object detectors without changing the target objects.

3.2 Adversarial Border

3.2.1 Methodology

The proposed adversarial examples are constructed using Faster R-CNN (Figure 3.1), which is composed of three main sub-networks: feature extraction network, region proposal network (RPN), and detection network. The feature extraction network is used to compute features consumed by the RPN and the detection network. It is also called a backbone network and VGG-16 and ResNet are commonly employed as the feature extraction network. The RPN takes features from the feature extraction network and determines a set of region proposals, which have a high probability of containing objects. The detection network takes both outputs from the feature extraction network and the RPN as input and computes final bounding

¹The work in this chapter has been published as “Attacking object detectors without changing the target object.” In Pacific Rim International Conference on Artificial Intelligence, 2019 and “Adversarial Signboard against Object Detector.” In BMVC, 2019.

FIGURE 3.1: Illustration of Faster R-CNN.

boxes and their corresponding classes using respectively the box regression network and the box classification network (see Figure 3.1). More precisely, the box regression network computes parameters to refine the coordinates of the input region proposals and determines the final bounding boxes and the box classification network produces a probability matrix P , each of whose row and column correspond respectively to one input region proposal and one specific category. The element in the i^{th} row and the j^{th} column in P indicates the probability of the i^{th} region proposal belonging to the j^{th} category.

Different from other adversarial examples against object detectors which make use of the classification layer, adversarial border exploits the regression layer. The regression layer is used to compute bounding box coordinates of objects. For each region proposal, the regression layer outputs four values d_x, d_y, d_w , and d_h to refine the region proposal's coordinates (p_x, p_y, p_w, p_h) , where (p_x, p_y) is the top left hand corner of the region proposal and p_w and p_h are its width and height, respectively. The bounding box coordinates are calculated by the following equations:

$$g_x = p_w \times d_x + p_x \quad (3.1)$$

FIGURE 3.2: Examples for (A) perturbed training images and (B) perturbed testing images.

$$g_y = p_h \times d_y + p_y \quad (3.2)$$

$$g_w = p_w \times e^{d_w} \quad (3.3)$$

$$g_h = p_h \times e^{d_h} \quad (3.4)$$

The equations imply that if d_w and d_h are slightly distorted, the bounding box coordinates will change a lot due to the exponential function. Adversarial border is designed to mislead the regression layer such that it outputs very large bounding boxes for the target object, e.g., stop sign, and finally decreases the output confidence to it. The size of the bounding box, i.e., $g_w \times g_h$ and the size of the corresponding region proposal, i.e., $p_h \times p_w$ are always similar. Faster R-CNN is designed in such a way so that the parameters d_w and d_h can be learned more effectively. Mathematically, when $g_w \approx p_w$ and $g_h \approx p_h$, it implies that $d_w \approx 0$ and $d_h \approx 0$. If the regression layer is misled to output $d_w = v$ and $d_h = v$, where $v > 0$, the size of the output bounding box will be roughly $e^v \times e^v$ times larger than

the size of the target object. In the experiments, v is set to 1, which corresponds roughly to 738% enlargement.

To handle the scale, distance, and lighting variations in the physical world, n images are sampled from a video collection V which contains a target object T as a training set. Let the adversarial border be a patch Λ with a fixed size of $s \times s$ pixels and the bounding box predicated by Faster R-CNN for the target object T in a training image I_i be b_{ti} with a size of $h_{ti} \times w_{ti}$ pixels. The patch Λ is first enlarged to $\alpha h_{ti} \times \beta w_{ti}$ pixels, where $\alpha > 1$ and $\beta > 1$. Then, the enlarged patch is inserted to I_i at the location b_{ti} . Note that the image pixels inside b_{ti} are retained and only the pixels outside b_{ti} are replaced with the pixels in the enlarged Λ . These operations are represented by $Z_{\alpha\beta}$ and $I_{pi} = Z_{\alpha\beta}(\Lambda, b_{ti}, I_i)$ represents the perturbed image. Figure 3.2 (A) shows some perturbed images in training. Note that in training, the adversarial border is inserted at the location of b_{ti} , which is determined by Faster R-CNN, and therefore, it may overlap with the target object. However, in testing and attacking, it is assumed that b_{ti} is not available, which is important for black-box attacks, and the ground truth bounding box of the target object is used to insert the trained adversarial border. Figure 3.2 (B) shows some testing images with a trained adversarial border. In fact, in preliminary experiments, the enlarged Λ was inserted in the ground truth location of the target object. However, its performance is not as good as the one inserted at the location of b_{ti} . Let $f_h(\cdot)$ and $f_w(\cdot)$ be two functions representing the operations in Faster R-CNN to compute the scaling values of d_w and d_h . Since the box-regression layer in Faster R-CNN outputs one boundary box for each region proposal and the number of region proposals is fixed for each image, f_h and f_w produce two fixed-length vectors, D_w and D_h , whose elements are respectively d_w and d_h of one bounding box and whose dimensions are same as the number of bounding boxes. Faster R-CNN uses non-maximum suppression and confidence values to determine final object bounding boxes. Mathematically, given a set of input images, to train

TABLE 3.1: The original detection rates of Clip 1 – Clip 5 for digital attacks

	Resolution	Number of stop signs	Faster R-CNN	YOLOv3
Clip1	1280 × 720	94	1.000	1.000
Clip2	1280 × 720	100	0.800	0.790
Clip3	1280 × 720	74	1.000	1.000
Clip4	1280 × 720	157	0.981	0.987
Clip5	406 × 720	298	0.997	0.876

an adversarial border, the minimization is performed through the equation:

$$\min_{\Lambda} \sum_{i=1}^n ||\mathbf{I} - f_w(Z_{\alpha\beta}(\Lambda, b_{ti}, I_i))||^2 + ||\mathbf{I} - f_h(Z_{\alpha\beta}(\Lambda, b_{ti}, I_i))||^2 \quad (3.5)$$

where $\mathbf{I}$ represents a vector all whose elements are one and its dimension is same as D_h and D_w .

3.2.2 Experimental Results

To evaluate adversarial border, stop sign is used as a target object because it was commonly used in the previous adversarial example studies and an important object for autonomous vehicles. In this evaluation, both white-box attacks and black-box attacks are studied. For white-box attacks, adversarial border is trained and evaluated on Faster R-CNN with VGG-16 as a backbone network, while for black-box attacks, it is also trained on Faster R-CNN with VGG-16 but examined on YOLOv3 [65]. Both Faster R-CNN and YOLOv3 were trained on the COCO dataset to detect multiple objects, including stop signs. In addition to white-box and black-box attacks, the performance of adversarial border for digital attacks and physical attacks is also investigated.

Digital Attack Evaluation

For evaluating adversarial border in digital attacks, five in-car videos with stop signs were downloaded from the Internet. Figure 3.4 shows sample frames from

FIGURE 3.3: The 12 adversarial borders used in the experiments. The adversarial borders in the 1st-4th columns are respectively computed from α_1 to α_4 and the adversarial borders in the 1st-3rd are respectively obtained at the 100th, 200th and 300th iterations.

the videos. In each of the videos, a car was approaching a stop sign. Since adjacent frames in the videos are similar, each alternative frame in the videos is sampled and five clips named Clip 1 to Clip 5 are constructed for evaluating adversarial border in digital attacks. All the images in these five clips have a stop sign facing the camera. Table 3.1 summarizes the total number of images in each clip and the corresponding detection rates obtained by Faster R-CNN and YOLOv3. Clip 1 is used to train adversarial border and the other four clips are used to test its impact on Faster R-CNN and YOLOv3. The size of adversarial border is controlled by β and α . In the experiment, $\alpha = \beta$ is set and four values of α computed by the equation,

FIGURE 3.4: Sample frames from the five videos for evaluating digital attack.

$$\alpha_i = 2 \times \left(\frac{i}{6}\right) + 1 \quad (3.6)$$

where $i \in \{1, 2, 3, 4\}$, are used to generate adversarial borders with four different sizes. Size 1 to Size 4 are used to denote the sizes of the adversarial borders computed from α_1 to α_4 , respectively. Since the loss decreases slowly between the 200th iteration and the 300th iteration, the optimization is terminated at the 300th iteration. The adversarial borders generated at the 100th, 200th and 300th

TABLE 3.2: Detection rates of Faster R-CNN and YOLOv3 for the videos collected in location 1 and location 2 with adversarial borders.

Video	Total frames	Faster R-CNN	YOLOv3
Location 1			
Left hand side view	181	0.796	0.751
Front view	194	0.743	0.557
Right hand side view	149	0.832	0.497
Mean	175	0.79	0.602
Location 2			
Left hand side view	107	0.813	0.542
Front view	106	0.887	0.774
Right hand side view	95	0.863	0.611
Mean	103	0.854	0.642

iterations are used in this experiment. The adversarial borders with different sizes and generated by different iterations are shown in Figure 3.3. The number of detected stop signs with and without adversarial borders in each clip are denoted as Det_{adv} and Det_{org} respectively. The successful attack rate for digital attacks is defined as:

$$AR = 1 - Det_{adv}/Det_{org} \quad (3.7)$$

Table 3.3 lists the average successful attack rates of Clip 2 to Clip 4 against Faster R-CNN and YOLOv3. It shows that the adversarial borders can successfully mislead Faster R-CNN and can also be transferred to YOLOv3 with high successful attack rates. In other words, adversarial border can be used in white-box and black-box attacks. Figure 3.6(A) shows the detection results with and without the adversarial examples.

Physical Attack Evaluation

TABLE 3.3: Average successful attack rates of digital attacks

	Faster R-CNN			YOLOv3		
Iteration	100	200	300	100	200	300
Size 1	0.626	0.790	0.826	0.626	0.847	0.885
Size 2	0.854	0.912	0.908	0.753	0.909	0.941
Size 3	0.882	0.857	0.819	0.884	0.871	0.904
Size 4	0.789	0.816	0.833	0.686	0.763	0.788

For evaluating adversarial border in physical attacks, one original stop sign and the 12 adversarial borders (Figure 3.3) with the original stop sign were printed on A3 paper using HP Colour LaserJet Enterprise flow MFP M880. The stop signs in these printouts were from the same image downloaded from Google image, not from the training video, Clip 1. The stop signs with the 12 different adversarial borders have the same size in the printouts but the sizes of the adversarial borders are different (Figure 3.5). Each of them was placed in two different outdoor locations and three videos from the left hand side, the front view and the right hand side were taken from it using Samsung Galaxy 7. Totally, 72 ($12 \times 3 \times 2$) videos were taken from the 12 stop signs with the adversarial borders and 6 ($1 \times 3 \times 2$) videos were taken from the original stop sign without adversarial border. The resolution of the videos is 1080 by 1920 pixels. As with the previous experiment, images are extracted from every alternative frame from these videos for evaluation. Table 3.2 lists the detection rates of the six videos collected from the original stop sign. It shows that Faster R-CNN and YOLOv3 can detect around 80% and 60% of the frames, respectively. The successful attack rate for digital attacks defined Eq. 3.7 cannot be employed to evaluate physical attacks because the videos with and without the adversarial borders are different, e.g., viewpoints and the number of frames. Instead, the Faster R-CNN and YOLOv3 detection rates for the stop signs with the adversarial borders are computed and their means are given in Table 3.4. Each value in Table 3.4 is the mean detection rate computed from three different videos collected at the same location. Comparing with the detection rates

FIGURE 3.5: An adversarial border and the original stop sign on printouts.

TABLE 3.4: Detection rates of Faster R-CNN and YOLOv3 for the videos collected in location 1 and location 2 with adversarial borders.

	Faster R-CNN			YOLOv3		
Iteration	100	200	300	100	200	300
Location 1						
Size 1	0.644	0.674	0.589	0.533	0.549	0.516
Size 2	0.421	0.532	0.675	0.393	0.385	0.506
Size 3	0.337	0.356	0.349	0.287	0.270	0.215
Size 4	0.441	0.413	0.488	0.296	0.301	0.282
Location 2						
Size 1	0.354	0.358	0.354	0.118	0.150	0.121
Size 2	0.035	0.118	0.248	0.108	0.107	0.145
Size 3	0.058	0.135	0.152	0.121	0.094	0.069
Size 4	0.300	0.301	0.266	0.144	0.115	0.093

with and without adversarial borders respectively in Table 3.4 and 3.2, it shows that adversarial border can effectively mislead Faster R-CNN and YOLOv3 in the physical world. Figure 3.6 (B) shows the detection results with and without the adversarial examples in the physical world.

FIGURE 3.6: Detection results on images with and without adversarial borders. (A) Results from the digital attacks and (B) results from the physical attacks.

3.3 Adversarial Signboard

3.3.1 Methodology

The adversarial border can fool object detectors digitally and physically. However, it makes the object looks weird. Adversarial signboard which looks like a signboard is designed to make the attack looks natural. Adversarial signboard aims at misleading the box classification network into outputting low probabilities for the target object, i.e., stop sign in this study. To enhance its robustness to scale, distance and lighting variations in the physical world, n images, each of which contains one target object T , are sampled from k videos $V_1, \dots, V_k$ as a training set. Given an input image I , let $B(I)$ be the bounding boxes outputted by the box regression network and $P_T(I)$ be the corresponding probabilities of the target object T . More clearly, $P_T(I)$ is the column of P corresponding to the class of the target object. Since not all the bounding boxes in $B(I)$ are generated by the target object, only the bounding box $b_t(I)$ with the highest probability in $P_T(I)$ is selected for training. It is assumed that in each training frame, there is one target object and Faster R-CNN can roughly locate it. To train adversarial signboard on a large database without ground truth bounding boxes, $b_t(I)$ is used as a reference to scale and place an adversarial signboard in I . Let the adversarial signboard be a patch Λ with a fixed size of $h \times w$ and the size of $b_t(I)$ be $h_t \times w_t$. The adversarial signboard is scaled to $\alpha h_t \times \beta w_t$. In the experiments, α and β are set to 1 and 2 respectively. If the bounding box given by Faster R-CNN is not accurate enough and if the adversarial signboard is placed right below $b_t(I)$, it will cover a part of the target object. To address this problem, the adversarial signboard is placed at a distance below $b_t(I)$. In the experiments, the distance D is set to $1/10 \times h_t$. Furthermore, it also makes the position of the adversarial signboard more natural. In physical attacks, it is hard to put the adversarial signboard in an accurate position as digital attacks. Therefore, random shifts in both vertical and horizontal directions are

added into the training process to improve its robustness against this inaccuracy. The range of the random shifts in the horizontal direction is from $-1/10 \times w_t$ to $1/10 \times w_t$ and the range of the random shifts in the vertical direction is from 0 to $1/10 \times h_t$. After the random shifts, the distance between the adversarial signboard and the target object varies from $1/10 \times h_t$ to $2/10 \times h_t$. To mimic real signboard, a text mask is applied to the adversarial signboard and only the pixels outside the text mask are involved in the training. Let $m(\Lambda)$ be the process of putting the text mask in the adversarial signboard and $f(I, m(\Lambda), b_t(I))$ be the process of scaling and placing the adversarial signboard with the text mask on I . Note that the output of $f(I, m(\Lambda), b_t(I))$ is an image with the adversarial signboard. Instead of minimizing the mean probability, the proposed objective function minimizes the expected maximum original class probability of all the bounding boxes in the video collections V , which can be written as:

$$\min_{I \in V_1, \dots, k} \mathbb{E}(\max(P_T(f(I, m(\Lambda), b_t(I)))))) \quad (3.8)$$

Though both adversarial border [66] and adversarial signboard are placed outside target object to perform attacks, they are derived by different algorithms. Adversarial border is designed to fool the box regression network, while adversarial signboard is designed to fool the box classification network. The former minimizes the mean difference between the bounding box parameters produced by the box regression network and predefined target values, while the latter minimizes the expected maximum original class probability. Furthermore, adversarial border has to surround target object tightly to perform attacks and makes it look weird (Figure 3.2). Adversarial signboard significantly lightens the posing requirement. By placing adversarial signboard below the target object with some distance, it is enough to perform an attack. Besides, it camouflages itself as a normal signboard, which is much more natural than an adversarial border. Figure 3.7 shows normal signboards and Figure 3.8 shows adversarial signboards with different text

FIGURE 3.7: Normal signboards collected from the Internet.

designs. Furthermore, in the previous study, adversarial border was only evaluated on Faster R-CNN with VGG-16 as a backbone network and YOLOv3, while in this study, adversarial signboard is evaluated on Faster R-CNN with VGG-16 and ResNet-101 as backbone networks, Mask R-CNN, YOLOv3 and R-FCN.

3.3.2 Experimental Results

Same as the experiment of adversarial border, stop sign is used as a target object to evaluate adversarial signboard. In the experiments, 529 images sampled from 4 videos and Faster R-CNN with ResNet-101 as a backbone network are used to train three adversarial signboards with a size of 90×180 pixels. The three adversarial signboards with different text masks shown in Figure 3.8 (A)-(C) are respectively denoted as AS-W, AS-B and AS-BW. In the experiments, both white-box and black-box attacks are studied. For white-box attacks, Faster R-CNN with ResNet-101 as its backbone network is used to evaluate the adversarial signboards. For black-box attacks, four object detectors YOLOv3, R-FCN, Faster R-CNN using VGG-16 as its backbone network and Mask R-CNN, are used to evaluate the adversarial signboards. The backbone networks of YOLOv3, R-FCN and Mask R-CNN are respectively Darknet-53, ResNet-101 and ResNet-101. For the sake of convenience, the two Faster-CNN detectors are denoted as Faster R-CNN(VGG-16)

FIGURE 3.8: The first column is the three adversarial signboards for digital attacks and the second column is their printouts for physical attacks.

and Faster R-CNN(ResNet-101). In addition to white-box and black-box attacks, they are also examined in digital and physical attacks.

Digital Attacks Evaluation

In digital attacks, 36 videos collected from the Internet are employed. Since adjacent frames in the videos are highly similar, each alternative frame is selected and forms a testing set with 4,073 images. The adversarial signboards are scaled and placed in the testing images. Figure 3.9 gives sample detection results with and without adversarial signboards. The successful attack rate AR for one video is defined as the number of successful attacks divided by the number of original detected images. Note that AR is computed from each video and the average AR from the 36 videos is used as a final performance index for digital attacks. Table

TABLE 3.5: The average successful attack rates in digital attacks (%)

	Faster R-CNN (ResNet-101)	Faster R-CNN (VGG-16)	Mask R-CNN	YOLOv3	R-FCN
DR-ORG	89	70	95	95	76
AS-B	87	58	75	50	59
AS-W	88	78	67	57	70
AS-BW	84	71	68	58	64
Mean AR	86	69	70	55	64

TABLE 3.6: The average detection rates in physical attacks (%)

	Faster R-CNN (ResNet-101)	Faster R-CNN (VGG-16)	Mask R-CNN	YOLOv3	R-FCN
Stop sign	98	80	94	68	96
AS-B	20	47	16	18	47
AS-W	30	43	18	13	38
AS-BW	32	57	19	16	59

3.5 lists the original detection rates (DR-ORG) and the average successful attack rates for white-box and black-box attacks. For white-box attacks, the adversarial signboards achieve average successful attack rates over 84%. For black-box attacks, they can mislead detectors with similar architectures, i.e., Faster R-CNN(VGG-16) and Mask R-CNN at high successful attack rates. The mean average ARs of these two detectors are 69% and 70%, respectively. For R-FCN, the average AR varies between 70% and 59%. For YOLOv3, the average AR varies between 58% and 50%. The experimental results also show that AS-W performs the best among the three adversarial signboards and its mean average AR across all the detectors is 72%. These experimental results demonstrate that the adversarial signboards can significantly reduce the number of detected stop signs in both white-box and black-box attacks.

Physical Attacks Evaluation

In this experiment, a stop sign with a size of 19.2 cm by 19.2 cm and the three adversarial signboards with a size of 19.2 cm by 38.4 cm are printed out. Figure 3.8

FIGURE 3.9: Detection results with and without adversarial signboards from Faster R-CNN(ResNet-101)

shows the stop sign and the adversarial signboards. Four groups of videos are taken in an open car-park by a smart-phone camera and each group has one video for the stop sign without adversarial signboard and three videos for the three adversarial signboards with the stop sign above. In each group, the videos are taken from roughly the same viewpoint, location and camera direction. Thus, in total, 16 videos with a resolution of 1920 by 1080 (or 1080 by 1920) pixels are collected and the videos in the same group are more comparable. On average, each video has 115 frames used in this experiment. In the videos, the stop sign is on the right hand side, which is the same as in the training videos. Figure 3.10 shows two sample frames from the videos. The average successful attack rate cannot be employed as a performance index in physical attacks, because the videos with and without adversarial signboards are not collected from the exact same viewpoint and at the same time and have different numbers of frames. Thus, the average detection rates given in Table 3.6 are used a performance index. All the detectors except for YOLOv3 perform well on the original stop sign with detection rates of over 80%. The adversarial signboards significantly reduce the detection rates of Faster R-CNN (ResNet-101), YOLOv3, and Mask R-CNN. All their detection rates are below 32%. Seven out of the nine detection rates from these detectors

FIGURE 3.10: Sample frames with and without adversarial signboard from the videos in the same group.

are even below 21%. The adversarial signboards also decrease the detection rates of R-FCN substantially. Its average detection rate is 48%. Comparing with the original detection rate, the adversarial signboards reduce the detection rates by 48%. The adversarial signboards have less impact on Faster R-CNN(VGG-16), but they still can reduce its average detection rate by 31%. As with the digital experiment, AS-W performs the best.

3.4 Summary

Object detector is an essential component in many computer vision and AI systems. The previous adversarial studies pointed out that by significantly changing pixels inside the target object, it is possible to fool object detectors. In this chapter, two new types of adversarial examples: adversarial border and adversarial signboard, which do not change any pixel in the target object, are proposed against Faster R-CNN. Adversarial border aims at misleading the regression network into outputting very large bounding boxes and finally decreases the output confidence to the target object. Adversarial border surrounds the target object tightly and makes it look weird. Adversarial signboard is proposed to solve this issue. It is placed below the target object at a distance and looks more natural. Extensive experiments show that the proposed adversarial examples can effectively fool object detectors digitally

and physically. They also demonstrate that adversarial border and adversarial signboard can be used to perform both white-box and black-box attacks

Chapter 4

New Threats against Object Detector with Non-Local Block

4.1 Introduction

Convolutional neural networks (CNNs) have become an essential component in computer vision systems where many of them have been deployed commercially. Traditional CNNs are built on local operators, such as convolution and pooling. In order to extract information in a wide area, the local operators are stacked, resulting in larger receptive fields in theory. However, Luo et al. pinpointed that the effective receptive fields of CNNs are much smaller than their theoretical receptive fields [67]. Furthermore, stacking the local operators, especially convolution, would dramatically increase the computational cost and cause optimization difficulties [46]. To address these issues, Wang et al. generalized the classical non-local mean operator for image denoising [68] and proposed non-local blocks [69] in 2018. Since the non-local blocks can be easily inserted into many existing architectures and combined with other operators, they have been taken as a generic family of building blocks in CNNs for capturing long-range dependencies. In fact, the non-local neural

network has raised a new trend in computer vision with an outstanding performance in object detection [70–73], as well as other various techniques on this basis, such as action recognition [74, 75] and person re-identification [76]. There are nearly 200 related articles published in top computer vision venues during the past two years. Non-local blocks have also been applied in other research fields, such as computer-assisted radiology and surgery, bioinformatics, electronic and automation control, speech processing, *etc.* With it being a heated topic, and its potentials, it has also drawn attention from industry and companies, including Facebook [69], Microsoft [77], Baidu [78], Tencent [79, 80], Huawei [81], Face++ [82], *etc.*

However, massively applying the non-local blocks in commercial systems without studying their threats would be risky. Traditional CNNs are well-known in suffering from adversarial examples. Researchers have demonstrated that using carefully crafted adversarial examples, they can mislead different CNNs designed for various computer vision problems. These include image and video classification, segmentation and object detection [5, 7, 63, 83–85]. To mislead object detectors, the current attacks either modify pixels inside target objects [6, 7, 86] or put the adversarial patches very close to target objects [84, 85]. Though non-local blocks have been applied to a lot of computer vision problems, object detection is selected for this study because it is an essential component in many cyber-physical systems, *e.g.*, autonomous vehicles. In order to investigate whether the non-local blocks would bring new threats to object detectors, two types of attacks are studied in this chapter¹ — appearing and disappearing attacks. The former aims at misleading the object detector such that it is unable to detect a target object category, *e.g.*, stop sign, and the latter aims at misleading the object detector such that it detects a predefined object category which is not present in images. If an adversarial patch is similar to a target object, the object detector would detect it as a target object. However, this is not the goal of appearing attack where wrongly detected target

¹The work in this chapter has been published as “New Threats against Object Detector with Non-Local Block”, ECCV, 2020

objects should appear beyond the adversarial patch itself. Different from the previous adversarial examples against object detectors, in these attacks, the adversarial patches are required neither to be put very close to the target objects [84, 85] nor to overlap with them [6, 7, 86].

To study the threats caused by non-local block, a non-local block is added into a Faster R-CNN and algorithms are designed to craft adversarial examples for carrying out these two types of attacks. By comparing the experimental results of the adversarial examples on the Faster R-CNN with non-local block and the original Faster R-CNN as a control, new threats from non-local blocks can be identified. To further explain the threats, the reception fields of non-local blocks are studied empirically and theoretically.

4.2 Methodology

4.2.1 Faster R-CNN and Non-local Block

To study the threats caused by the non-local blocks, adversarial patches are designed for carrying out disappearing and appearing attacks. In this study, Faster R-CNN with a ResNet-101 as its backbone and a non-local block is used to train the adversarial patches. The Faster R-CNN is selected on the account of its popularity and that many detectors are relying on the Faster R-CNN architecture. For a clear presentation, the original Faster R-CNN without the non-local block is first described. It consists of three major components: a backbone network, a region proposal network (RPN) and a detection network. The backbone network computes features for both RPN and detection network. The RPN takes the features and produces region proposals that have high probability with objects. The detection network takes the features and the region proposals as inputs. Its box regression layer refines the bounding box coordinates provided by RPN and its

classification layer outputs a probability matrix, P , each of whose row and column respectively corresponds to one region proposal and one class label. The element in the i^{th} row and the j^{th} column of P representing the probability of the i^{th} region proposal belonging to the j^{th} class is denoted as p_{ij} and the i^{th} row and the j^{th} column of P are denoted $p_{i\cdot}$ and $p_{\cdot j}$ respectively. Faster R-CNN applies a threshold and non-maximum suppression to determine final object classes and their bounding boxes.

For this study, a non-local block is inserted between the 3^{rd} and 4^{th} residual blocks in the ResNet-101. It is inserted at the same location as Wang et al. [69], and inserting at a lower layers would require a lot of memory and computation power when training a non-local block. For the sake of convenience, the term Faster R-CNN-WN is used to refer to the Faster R-CNN with the non-local block. Several types of non-local blocks have been proposed. The embedded Gaussian non-local block is employed in this study because different types of non-local blocks have very similar effect [69] and embedded Gaussian non-local block is the most popular one among them. Formally, a non-local block is defined as

$$z_k = W_z y_k + x_k \quad (4.1)$$

where x_k is a feature vector at the k^{th} spatial location of the previous layer, W_z is a matrix optimized through training and

$$y_k = \frac{1}{C(x)} \sum_{\forall m} f(x_k, x_m) g(x_m). \quad (4.2)$$

where m is the index that enumerate all possible feature vectors of the previous layer.

In embedded Gaussian non-local block,

$$f(x_k, x_m) = e^{\theta(x_k)^T \phi(x_m)} \quad (4.3)$$

where $\theta(x_k) = W_\theta x_k$, $\phi(x_m) = W_\phi x_m$, the normalization factor is set as $C(x) = \sum_{\forall m} f(x_k)f(x_m)$ and $g(x_m) = W_g x_m$.

4.2.2 Disappearing Attack

Let T be a target object category, I be a training image and $B_{gt.T}(I)$ be the ground truth bounding box of a target object in I with a size of $w_{gt.T} \times h_{gt.T}$ pixels. To carry out disappearing attack, an adversarial patch Λ with a size of $w \times h$ pixels is constructed to minimize all the probabilities of the target object category in P , i.e., $p_{.T}$. Let $p_{jT} = F_N(I, j)$, where F_N represents the operations in the Faster R-CNN-WN computing the probability of the j^{th} region proposal belonging to the target class. To enhance the robustness of the adversarial patch for target objects in different images taken from different viewpoints and illumination conditions, Λ is trained on images from k videos, each of which contains at least one target object. The entire training set is denoted as Q . To properly model the variations of zoom factors and the distance between camera and target object in different images, Λ is resized according to the target object. More precisely, Λ is resized to $\alpha w_{gt.T} \times \beta h_{gt.T}$, where α and β are parameters controlling the size of Λ in the image. In the training, Λ is placed below $B_{gt.T}(I)$ at a distance. If the training image has more than one target object, one of them is randomly selected and Λ is placed below it. Though the relative location between Λ and the target object is fixed in training, in testing, Λ can be placed in different locations to perform the attack. Section 4.4 will explain why the difference between training and testing locations is not important. Let $f(I, \Lambda, B_{gt.T}(I))$ be a training image with the rescaled adversarial patch. In the training, Λ is trained to minimize the sum of $p_{jT}, \forall j$. The objective function

$$\underset{\Lambda}{\operatorname{argmin}} \sum_{I \in Q} \sum_{j \in \phi^C} F_N(f(I, \Lambda, B_{gt.T}(I)), j) \quad (4.4)$$

is used to perform this minimization.

4.2.3 Appearing Attack

In this sub-section, we use the same notations as in the previous sub-section. As with the previous attack, an adversarial patch Λ with a size of $w \times h$ is inserted into a training image I , according to the location of a reference object and its size. The size of the rescaled Λ is set to $\alpha w_{ht} \times \beta h_{ht}$ pixels and Λ is placed below the reference object at a distance. In the experiments, stop sign is used as a reference object for placing and rescaling Λ in the appearing attack. The image with the rescaled Λ is denoted as $f(I, \Lambda, B_{gt.T}(I))$. To carry out this attack, a target label T is selected and the adversarial patch Λ is trained to minimize the negative $\log p_{jT}$ for the region proposal not overlapping with Λ . To avoid the appearing objects overlapping with Λ , the objective function also minimizes the negative $\log p_{jB}$ for region proposals overlapping with Λ , where p_{jB} is the background probability of the j^{th} region proposals. Mathematically, the objective function below is used to train Λ ,

$$\begin{aligned} \underset{\Lambda}{\operatorname{argmin}} \sum_{I \in Q} \sum_{j \in \phi^C} -\varepsilon \log F_N(f(I, \Lambda, B_{gt.T}(I)), j) \\ - \sum_{j \in \phi} (1 - \varepsilon) \log F_{NB}(f(I, \Lambda, B_{gt.T}(I)), j) \end{aligned} \quad (4.5)$$

where ε is a parameter balancing the two terms, F_{NB} represents the operations of the Faster R-CNN-WN computing p_{jB} , ϕ is a set storing indices of the region proposals overlapping with Λ and ϕ^C is its complement storing indices of the region proposals not overlapping with Λ .

In this objective function, the first term is to mislead the region proposals that are not intersecting with Λ and make them target objects. The second term is designed to keep the proposals intersecting with Λ to be the background. It is noticed that

FIGURE 4.1: Adversarial patches for the (A) disappearing attacks and (B) appearing attacks against the Faster R-CNN-WN. Adversarial patches for the (C) disappearing attacks and (D) appearing attacks against the Faster R-CNN.

the appearing attack will be weak when ε is too small, and bounding boxes will appear around or inside Λ when ε is too large (Figure 4.3 (C)).

4.3 Experiments

To evaluate the threats caused by non-local block, appearing and disappearing attacks are performed on the Faster R-CNN with the non-local block (Faster R-CNN-WN). The attacks are also performed on the original Faster R-CNN [87] without the non-local block, as a control experiment. The two Faster R-CNNs are trained on the COCO dataset (2017 training images) and their backbone networks are ResNet-101. In the disappearing attacks, stop sign is selected as a target object and also a reference object for resizing and placing the adversarial patches in the images. In the appearing attacks, stop sign is selected as a reference object, because it is a common target object in the previous adversarial example studies [6, 7, 84, 86] and an important object for autonomous vehicles. In the appearing attacks, boat is selected as a target object because none of the training and testing videos has boat and it would make the evaluation easier. In the experiments, 721 images sampled every other frame from five in-car videos are used as a training set. The image sizes are 406×720 pixels or 1080×720 pixels. Four adversarial patches with a size of 200×200 pixels are generated. Figure 4.1 shows the four adversarial patches. These adversarial patches are used to evaluate the threats caused by the

non-local block in both digital and physical worlds. More experiment results on Open Image Dataset are provided in the section 4.5.

4.3.1 Digital Attack

In digital attack, 4073 images sampled from 36 Internet in-car videos are taken as a testing set. The sizes of the images are 1080×1920 pixels and the sizes of the stop signs range from 21×22 pixels to 660×633 pixels. The original detection rates of the Faster R-CNN-WN and the Faster R-CNN are 80.6% and 78.6%, respectively. These results match with Wang et al.'s findings [69] that non-local block can improve detection performance. The adversarial patches are scaled and then placed below the detected stop signs with a certain distance away from it. In theory, the attack rate would be higher with a larger adversarial patch. In the disappearing attack experiments, we would like to keep the size of adversarial patch similar to that of the traffic signs in the real world. Thus, α and β are set to 1.5. When at least one detectable stop sign in an original image is missing due to the adversarial patches, the attack is considered a success. Denote the number of original images with at least one detectable stop sign as Det_{imgs} . The successful disappearing attack rate D_{ar} defined as:

$$D_{ar} = \frac{\text{number of successful attacks}}{Det_{imgs}} \quad (4.6)$$

is used as a performance index. Table 4.1 lists the original detection rate (DR_{org}) and the successful disappearing attack rates of the two detectors. It indicates that the impact of the disappearing attacks is much greater on the Faster R-CNN-WN than the original Faster R-CNN. Figure 4.2 shows typical detection results from these two detectors.

In the appearing attack experiments, boat and stop sign are respectively selected as a target object and a reference object. Similar to disappearing attack, α and β

FIGURE 4.2: Disappearing attack results. (A) The detection results of stop sign from the original Faster R-CNN and (B) the detection results of stop sign from the Faster R-CNN-WN. The first column is the original detection results without the adversarial patches and the second column is the detection results with the adversarial patches.

TABLE 4.1: Successful disappearing attack rates (%)

	Original Faster R-CNN	Faster R-CNN-WN
DR_{org}	78.6	80.6
D_{ar}	7.6	52.7

are set to 2 and ε in Eq. 4.5 is set to 0.4. Note that without the adversarial patches, the detectors would still have the probability of wrongly detecting other objects or background as a boat. When the detector has wrong detections in absence of the adversarial patches, only the case where the intersection over union of these detected boxes and the predictions for boats is smaller than 0.6 will be considered as a result of the appearing attack. The adversarial patches may cause the detectors to wrongly detect multiple boats. The wrongly detected boats with no intersection with the adversarial patches are considered as a clear success (Figure 4.3 (A)). In some cases, the wrongly detected boats are very large and have some overlap with the adversarial patches (Figure 4.3 (B)). Hence, the intersection of the detected boat and the adversarial patch over the detected boat (IOD) is used to define successful attack. If there is no intersection, IOD will be zero and if the detected boat is completely inside the adversarial patch, IOD will be one. When IOD is smaller than a threshold, it is considered as a successful attack. The successful appearing attack rate A_{ar} is defined as

$$A_{ar} = \frac{\text{number of images with at least one successful attack}}{\text{number of testing images}} \quad (4.7)$$

Table 4.2 gives the successful appearing attack rates from the two detectors and Figure 4.3 shows some typical detection results from Faster R-CNN-WN and original Faster R-CNN. The Faster R-CNN can only detect the adversarial patch or its sub-region as the target object (Figure 4.3 (C)). However, the Faster R-CNN-WN is misled by the adversarial patch and detects large areas as boat. Since the original Faster R-CNN is insensitive to the disappearing and appearing attacks in the

TABLE 4.2: The successful appearing attack rates (%) under different IOD thresholds.

	Original Faster R-CNN	Faster R-CNN-WN
IOD=0	0	53.5
IOD<0.1	0.6	55
IOD<0.2	1.4	55.2
IOD<0.3	3.1	55.4
IOD<0.4	6.6	55.4

digital world, it is not included in the following physical experiments.

4.3.2 Physical Attack

To examine the threats caused by non-local block in the physical world, a stop sign with a size of 19.3 cm by 19.3 cm and the adversarial patches generated for disappearing and appearing attacks with a size of 28.5 cm by 28.5 cm are printed out. In the disappearing attack, 6 groups of videos with resolution of 1080×1920 or 1920×1080 pixels are taken from outdoor environments by a smartphone camera. In each group, there are 2 videos taken from the same location and roughly the same viewpoint. One video has the stop sign only; one video has the stop sign and the adversarial patch for disappearing attack. The adversarial patch is placed below or in front of the stop sign without a fixed distance (Figure 4.4 (B)). On average, 110 frames are sampled from each video for testing. As in the digital experiment, every other frame is sampled. In the digital disappearing attacks, the successful disappearing attack rate is used as a performance index. However, it is not applicable to physical attack because the videos with and without adversarial patches are neither collected from the exact same viewpoint nor at the same time, and have different numbers of frames. Thus, the detection rates with and without the influence from the adversarial patch are used to evaluate the impact and provided in Table 4.3. In some videos, there is also a real stop sign,

FIGURE 4.3: Appearing attack results, where boat is a target object. (A)-(B) The detection results of boat from the Faster R-CNN-WN. (C) The detection results of boat from the original Faster R-CNN.

so the detection rate here is defined as the number of detected stop signs divided by the number of stop signs in the images. Table 4.3 shows that for the Faster R-CNN-WN, the adversarial patch reduces the average detection rates from 87.2% to 48.0%. Figure 4.4 shows some example results of the physical disappearing attack.

In the digital appearing attacks, stop sign is just used as a reference object for resizing and placing the adversarial patch. Since it is impossible to resize the adversarial patch in physical attack and in fact, it is not necessary, stop sign is not used as a reference object. For examining appearing attacks in the physical world,

FIGURE 4.4: The detection results of stop sign from the Faster R-CNN-WN (A) without and (B) with the adversarial patch in disappearing attack. Note that the stop sign closed to the tree is a real stop sign.

4 groups of videos, with resolution of 1080×1920 pixels are taken in 4 locations. In each group, there are 2 videos with and without the adversarial patch for appearing attack taken from the same location and roughly the same viewpoint. The videos are sampled in every other frame. On average, 118 frames per video are used for testing. The appearing rate here is defined as the number of frames detected with the target object divided by the total number of the frames. Table 4.4 shows that for the Faster R-CNN-WN, the adversarial patch increases the average appearing rates of boat from 0.8% to 31.3%. Figure 4.4 shows some example results from the physical appearing attack.

4.3.3 Effective Attack Regions

In the previous disappearing attack experiments, the relative locations between the target object i.e., stop sign and the adversarial patches are fixed. More precisely, in both training and testing of the digital attacks, the adversarial patches are placed $1.5 h_t$ pixels below the stop sign, where h_t is the detected height of the stop sign in the image. In this experiment, the adversarial patches trained in the

FIGURE 4.5: The Faster R-CNN-WN detection results of boat in appearing attack.

FIGURE 4.6: Two sample images with the grids.

fixed relative position are placed in different locations. 831 images are sampled from every 5 frames from the 36 testing videos in the digital experiments. From them, 670 and 645 images are detected with stop sign by the Faster R-CNN-WN and the Faster R-CNN, respectively. Only the images with detected stop signs are employed in the following evaluation. Each image is divided by a grid and the size of each block in the grid is $1.5h_t \times 1.5w_t$. Note that the number of blocks in different images is different. Figure 4.6 shows images with the grids. In total 11×11 positions are tested. The adversarial patches are rescaled based on the stop signs and put in the centers of the blocks. If the detector cannot detect the stop sign, it is considered as a successful attack. Note that the size and the location of

TABLE 4.3: The detection rate of stop sign in disappearing attacks in the physical world.

Detection rate	Faster R-CNN-WN
Without attack	89.9
With attack	47.6

TABLE 4.4: The appearing rate of boat in appearing attacks in the physical world.

Appearing rate	Faster R-CNN-WN
Without attack	0.8
With attack	31.3

the stop sign in each image are different. To compute the successful attack rates at different relative locations, the grids are rescaled and aligned. Figure 4.7 shows the successful attack rates from both Faster R-CNN-WN and Faster R-CNN. The red boxes indicate the locations of the aligned stop signs. Note that the successful attack rates at different locations are computed from different numbers of images, because some stop signs are close to image boundary. Figure 4.7 (A) shows that the effective attack region for the Faster R-CNN-WN covers the entire grid and most of the successful attack rates are higher than 30%. The region close to the target object has much higher successful attack rates and the upper, lower and right borders have relatively lower successful attack rates. The rest of the region has similar successful attack rates. Without the non-local block, the successful attack rates become very low (Figure 4.7 (B)), except for the region very close to the center. These results indicate that disappearing attacks can be applied in long-distance and the difference between training and testing location is not a matter.

FIGURE 4.7: Successful attack rates in different locations, (A) Faster R-CNN-WN and (B) Faster R-CNN. The red boxes indicate the locations of the stop signs. The yellow lines indicate successful attack rates greater than 10% and the green line in (A) indicate successful attack rate greater than 30%.

4.4 Analysis of the Effective Attack Regions

The experiments in Section 4.3 expose several properties of the effective attack regions of the two detectors. To discuss the properties observed in Figure 4.7 systematically, in this section, an effective attack region is defined as where the successful attack rate is higher than 10%, which is highlighted by the yellow color boundaries. To understand these properties, analysis and discussion are provided in this section. The effective attack region of the Faster R-CNN is first discussed and then an analysis for the effective attack region of the Faster R-CNN-WN is provided. Since they are in fact the same, except for the non-local block in the backbone networks, the analysis below focuses on the reception fields of the backbone networks.

The effective attack region of the Faster R-CNN is very small and very concentrated on the center. Luo et al. [67] points out that the output neurons of deep CNNs, like ResNet-101, have a small effective receptive field with a Gaussian shape, which is much smaller than its theoretical receptive field. Thus, the adversarial patch can only affect its surrounding region and the successful attack rate decays exponentially when the distance between the target object and the adversarial patch increases (see Figure 4.7 (B)). Also because of the Gaussian shape effective receptive field, Figure 4.7 (B) has a roughly circular shape.

The effective attack region of the Faster R-CNN-WN (Figure 4.7 (A)) is very different. It covers the entire Figure 4.7 (A) with high successful attack rates. The region around the center has a very high successful attack rate and the successful attack rate in the rest of the region is roughly the same, except for the boundaries. It implies that the non-local block extends the effective receptive field to entire images. We would like to mathematically quantify how the receptive field has been changed after adding the non-local block on top of the convolution neural network. Here we only consider single-channel case for every layer and dot-product form of non-local block for simplicity. The result can be derived similarly for multi-channel cases by considering inter-channel correlations, and extended to other versions of non-local block by replacing the pairwise function f .

Following Luo et al.'s effective receptive field analysis [67], we compute the gradient of a single output neuron with respect to all input pixels. What we are interested here is the gradient of a single output pixel with respect to all input pixels, *i.e.*, $\partial y_i / \partial a_m$ for all $m \in \{1, \dots, N^2\}$, where y is the output of non-local block, and a is an $N \times N$ input image. The output of the CNN block, also the input of non-local block, is denoted by x . Without the loss of generality, we assume that x_i 's are uniformly distributed. The gradient can be decomposed to $\partial y_i / \partial a_m = \sum_k \partial y_i / \partial x_k \cdot \partial x_k / \partial a_m$ using chain rule, which is further divided into the following two cases,

$$\frac{\partial y_i}{\partial a_m} = \begin{cases} \sum_{k \neq i} \frac{\partial y_i}{\partial x_k} \cdot \frac{\partial x_k}{\partial a_m} + \frac{\partial y_i}{\partial x_i} \cdot \frac{\partial x_i}{\partial a_i}, & m = i \\ \sum_{k \neq m} \frac{\partial y_i}{\partial x_k} \cdot \frac{\partial x_k}{\partial a_m} + \frac{\partial y_i}{\partial x_m} \frac{\partial x_m}{\partial a_m}, & m \neq i \end{cases} \quad (4.8)$$

Recall that the dot-product version non-local operator follows the function $z_i = W_z y_i + x_i$, where

$$y_i = \frac{1}{\mathcal{C}(x)} \sum_{\forall j} f(x_i, x_j) g(x_j) = \frac{1}{N} \sum_{\forall j} x_i x_j W_g x_j \quad (4.9)$$

Thus, the gradient signal on x , $\partial y_i / \partial x_m$, can also be computed by considering the cases when $m = i$ and $m \neq i$,

$$\frac{\partial y_i}{\partial x_m} = \frac{1}{N} W_g \times \begin{cases} \sum_{j \neq m} x_j^2 + 3x_m^2, & m = i \\ 2x_i x_m, & m \neq i \end{cases} \quad (4.10)$$

It is presented by the previous work [67] that $\partial x_k / \partial a_m$ for all m forms a Gaussian shape, which diminishes fast at its tails. Thus, the summation terms in both cases of Eq. 4.8 are negligible since $\partial x_k / \partial a_m$ is close to zero when $|k - m|$ becomes larger. Moreover, we noticed in Eq. 4.10 that the gradient when $m = i$ is strictly greater than that when $m \neq i$ since $\sum_{j \neq m} x_j^2 + 3x_m^2 - 2x_i x_m$ is strictly positive, when not all x_i 's are zeros, and $\partial x_i / \partial a_i$ for all i is the value of a Gaussian density function taken at its mean, which is assumed to be equal for all i . We then conclude that the gradient $\partial y_i / \partial a_m$ when $m = i$ is greater than that when $m \neq i$. Besides, since not all x_i 's are zeros, the gradients $\partial y_i / \partial a_m$ is also non-zeros for all $m \neq i$.

This gives us an intuition of analyzing the receptive field by considering the non-local block. Different from the effective receptive field of CNNs without non-local block, which has non-zero gradient values in the neighborhood of the center pixel and zero gradients further away, the receptive field of CNNs with non-local block covers the whole input space. More specifically, the gradient of a single output pixel with respect to the corresponding pixel, $\partial y_i / \partial a_m$, is non-zero everywhere. It also proved that the gradient with respect to center pixel ($\partial y_i / \partial a_i$) is the maximum. With the consideration of the skip connection in the non-local block, which is the sum of x_j over all j in computing z_i , the neighborhood region of the center would have higher gradients because of its Gaussian shape receptive field [67]. The gradients of other regions are strictly smaller and have similar values.

The analysis above does not explain why the successful attack rate is lower at the boundary. When the adversarial patch is put in the image border, it would affect

lesser number of neurons in the input layer of the non-local block, x . Therefore, the attack would be weaker as illustrated in Figure 4.7. The experimental results also show that the adversarial patch trained at a fixed relative location with respect to the target object is effective in other locations. Because CNNs without a fully connected layer is roughly translation invariant, putting the adversarial patch in two different locations, p and q , their corresponding neuron outputs x_p and x_q should be roughly the same. Since the non-local block (Eq. 4.1) considers all $x_i, \forall i$, the adversarial patch can attack on other locations, different from the training location.

4.5 Extra Experiments on OpenImage Dataset

To further evaluate the threats caused by non-local block, appearing and disappearing attacks are performed on the Faster R-CNN with the non-local block (Faster R-CNN-WN) on Open Images Dataset V4. The object detector is trained on COCO dataset with 80 classes of objects. 24 classes overlapping with the object classes in Open Images Dataset V4 are selected for the following experiments.

4.5.1 Disappearing Attack

Different from the stop sign dataset, which normally just has 1 to 2 stop signs in each image, images in Open Images Dataset V4 usually have a lot of target objects. The adversarial patch has to be placed with a certain distance to all the target objects. The procedure to place Λ is given in algorithm 2. λ is used to control the distance between Λ and the target object used as a reference position and δ is used to control the minimum distance between Λ and other target objects. Images which do not have enough space to place the adversarial patch are not used in this evaluation. All the target classes have at least 100 valid images for this

evaluation. α and β are both set as 1.5. δ and λ are set as 1 and 0.5 respectively. For classes with more than 2000 images, only 2000 images are randomly sampled for the experiment. The images in each class are split equally as training and testing sets. Table 4.5. lists the results for disappearing attack. The average attack rate is 69.9%. Figure 4.8 shows some resultant images of disappearing attack.

4.5.2 Appearing Attack

The same 24 object classes are used in appearing attack. For each of the class, its appearing attack can be set as any 1 of the other 79 classes. In total, there are 1896 experiments. To reduce the number of experiments, we perform non-target appearing attacks. It means that if a newly detected bounding box which has not being detected before the attack, and has an IOU with the ground truth bounding box less than 0.7, then it is considered as a success, regardless of its class label. The objective function, which is similar as disappearing attack, aims at minimizing the scores for all the background bounding boxes and the scores for all the corresponding class bounding boxes. Thus, objects would be detected in the original background and other objects would be detected at the locations of the corresponding class objects. In this experiment, α and β are both set as 2. Table 4.6 lists the experimental results and some resultant images are given in Figure 4.9.

4.6 Summary

To overcome the weaknesses of traditional deep neural networks, which are ineffective to capture long-range dependency, researchers developed non-local blocks and demonstrated their effectiveness on various computer vision tasks. However, without understanding the threats caused by the non-local blocks and applying

FIGURE 4.8: Disappearing attack results. The first column is the original detection results without the adversarial patches and the second column is the detection results with the adversarial patches.

FIGURE 4.9: Appearing attack results. The first column is the original detection results without the adversarial patches and the second column is the detection results with the adversarial patches.

Algorithm 2: Procedure to add Λ to image

Input: I : original image; $b_{1,...,n}$: ground truth bounding boxes for target objects, n is the number of target objects; Λ : adversarial patch;

Output: a flag indicates if I is available and the position of Λ in I

```

1 set the flag to be false and the position of  $\Lambda$  to be empty ;
2 shuffle all the bounding boxes; compute the mean height  $h_m$  and width  $w_m$  of
  all  $b_i, i \in 1, \dots, n$ ; resize  $\Lambda$  to  $\alpha h_m \times \beta w_m$  ;
3 for each bounding box  $b_i$  in  $b_{1,...,n}$  do
4   place  $\Lambda$  below  $b_i$  with distance  $\delta h_i$ 
5   if  $\Lambda$  is inside the image and  $D_j/(h_j + w_j) > \lambda$ , where  $D_j$  is the distance
     between  $\Lambda$  and the bounding box  $b_j, j \in 1, \dots, n, j \neq i$  then
6     set the flag to be true and save the current position of  $\Lambda$  ;
7     goto final ;
8   place  $\Lambda$  above  $b_i$  with distance  $\delta h_i$ 
9   if  $\Lambda$  is inside the image and  $D_j/(h_j + w_j) > \lambda$ , where  $D_j$  is the distance
     between  $\Lambda$  and the bounding box  $b_j, j \in 1, \dots, n, j \neq i$  then
10    set the flag to be true and save the current position of  $\Lambda$  ;
11    goto final ;
12  place  $\Lambda$  on the left side of  $b_i$  with distance  $\delta w_i$ 
13  if  $\Lambda$  is inside the image and  $D_j/(h_j + w_j) > \lambda$ , where  $D_j$  is the distance
     between  $\Lambda$  and the bounding box  $b_j, j \in 1, \dots, n, j \neq i$  then
14    set the flag to be true and save the current position of  $\Lambda$  ;
15    goto final ;
16  place  $\Lambda$  on the right side  $b_i$  with distance  $\delta w_i$ 
17  if  $\Lambda$  is inside the image and  $D_j/(h_j + w_j) > \lambda$ , where  $D_j$  is the distance
     between  $\Lambda$  and the bounding box  $b_j, j \in 1, \dots, n, j \neq i$  then
18    set the flag to be true and save the current position of  $\Lambda$  ;
19    goto final ;
20 there is no suitable position to place  $\Lambda$ , set the flag to be false, the position of
    $\Lambda$  to be 0 ;
21 final ;
22 return the flag and  $\Lambda$  position;

```

them to critical systems is risky. In this chapter, two types of attacks, disappearing and appearing attacks against object detectors are studied. Different from the previous attacks against object detectors, these attacks are performed in long distance. The digital and physical experimental results show that the universal adversarial patches obtained by the proposed algorithms can mislead the Faster R-CNN with a non-local block to classify stop sign as background and to wrongly detect boats that are not in the images. To understand the effective attack region

TABLE 4.5: The successful disappearing attack rates (%) for different classes.

class	Airplane	Bench	Bottle	Bowl	Bicycle	Bird
detection rate	81.3	76.3	81.7	78.4	71.6	73.4
attack rate	83.2	79.1	61	60.7	55.7	55.9
class	Boat	Bus	Chair	Cake	Car	Cat
detection rate	62.6	83.5	77.5	70.4	79.4	74.1
attack rate	57.5	75.9	80.3	82.6	53.4	96.4
class	Clock	Dog	Fork	Horse	Motorcycle	Person
detection rate	68.0	80.6	71.3	84.1	76.9	78.3
attack rate	18.6	99.3	66.1	89.7	68.3	71.2
class	Tie	Train	Truck	Umbrella	Vase	Wine Glass
detection rate	81.6	67.1	73.3	69.4	78.4	83.0
attack rate	31.6	97	90	95.3	71.6	37.3

TABLE 4.6: The successful disappearing attack rates (%) for different classes.

class	Airplane	Bench	Bottle	Bowl	Bicycle	Bird
attack rate	40	29.8	46.1	15.5	34.8	15.3
class	Boat	Bus	Chair	Cake	Car	Cat
attack rate	12	17.6	57	33.3	26.9	28.2
class	Clock	Dog	Fork	Horse	Motorcycle	Person
attack rate	21.1	27.2	16.1	14.1	38	26.7
class	Tie	Train	Truck	Umbrella	Vase	Wine Glass
attack rate	24.7	21.5	25	21.4	17.2	50.6

and its properties, the reception field of the non-local block is analysed.

Chapter 5

From the Perceptive of CNN to Adversarial Iris Image

5.1 Introduction

IrisCode has been used to gather iris data for over one billion people from approximately 170 nationalities for national identity management, border control and various security applications [88–90]. The Government of India through the Unique Identification Authority of India has used IrisCode to register 1.1 billion citizens of India and is going to register all 1.2 billion citizens in the coming years for her national identity programme[91, 92]. Indonesia and Mexico also have similar national identity programmes based on IrisCode to register their full populations of 172 and 110 million citizens, respectively [93]. In addition to national identity, the U.S. Department of Defense has used IrisCode to enroll 10 million service personnel in a military electronic identity card scheme. The algorithm has also been used to deliver aid to refugees from the conflict in Syria [94] and for border controls, including 32 air, land and sea ports in the United Arab Emirates [89]. Because of its impact, researchers have developed various coding methods for biometric

recognition and template protection [89, 95–102] and also studied its theoretical and statistical properties [102–108].

Though enhancing the performance of IrisCode is important, studying its security threats is nothing less [23, 97, 106, 109, 110]. Kong indicated the existence of security risks within two patented template protection schemes that were deployed in commercial iris recognition systems for producing application-specific templates. These protected templates can be unlocked without the encryption key [105, 111]. Venugopalan and Savvides borrowed the idea from information hiding to embed iris information from a given iris template into a real iris image [23]. In their study, the given template was generated by the imaginary part of a one-dimensional log Gabor filter with only one scale and each pixel in their normalized iris image produced one bit information in the template. Their experimental results showed that the generated images and the given template can be matched with other iris images. However, only several generated images were given in the paper and no image quality assessment was performed. Galbally et al. used a genetic algorithm and matching scores returned by an iris recognition system to generate iris images [110]. The system used a coding scheme to construct iris templates, same as IrisCode but its filtering process was different. Galbally et al.'s method was in fact a hill climbing attack [112] and required unlimited access of the matcher, which returns numerical scores instead of binary decision. The iris templates employed in their study were composed of 9600 bits computed by one dimensional log-Gabor filters. The iris templates used in Venugopalan and Savvides's study and Galbally et al.'s study are significantly different from the templates of IrisCode, which are composed of 2048 phase bits and 2048 mask bits. The phase bits are extracted by two-dimensional Gabor filters with different scales and each mask bit corresponding to one phase bit at the same spatial location indicates whether the phase bit is corrupted by noise, such as eyelids or eyelashes. IrisCode does not generate phase bits for every pixel in the normalized iris images. Kong attempted to

generate iris images from templates of IrisCode [107]. He showed that IrisCode is in fact a compression algorithm with an extremely high compression ratio of 650 and derived its compression equation whose coefficients corresponding to the Gabor filters are quantized to one bit. Some basis vectors in the decompression equation are undefined. He used a graph-based method to estimate the original values of the quantized coefficients in the compression equation and PCA to fill the undefined basis vectors. He performed optimization in the frequency domain to determine the coefficients of the PCA basis vectors and to smooth generated images. Experimental results showed that the generated iris images and real iris images can be matched by IrisCode. However, their visual quality is still low, similar to JPEG images with the quality factor of 10. In another study, Kong found that templates of IrisCode can be considered as convex polyhedral cones [103]. Based on this property, he derived an extremely simple method, which can be implemented by one-line Matlab command, to generate iris images from the templates of IrisCode. As with the previous work, the generated iris images and real iris images can be matched by IrisCode. The author stated clearly that his work did not aim to generate high quality images from the templates. According to the best knowledge of the authors, no one successfully generates high quality iris images from the templates of IrisCode. In addition to generating iris images from templates, researchers have attempted to generate fingerprint images from the standard fingerprint templates [113, 114] and face images from templates of deep neural networks[115]. These generated images can be considered as adversarial examples, which aim to mislead machine learning and classification algorithms. Though a number of adversarial example generation methods have been developed recently for generic images, they are not applicable to IrisCode as they are designed for deep neural networks requiring differentiable objective functions. Designing and defending adversarial examples is a hot topic [116]. Because of the importance of

IrisCode and the flaws in the previous studies, in this ¹, a new algorithm based on constrained convex minimization with an inputted iris image selected by a search function as an information carrier is proposed.

5.2 A Computation Summary of IrisCode and Notations

Figure 5.1 illustrates the computational process of IrisCode. It first locates pupil, limbus, upper and lower eyelids and detects specular reflection. Then, it segments the iris and transforms it from the Cartesian coordinate to the polar coordinate. The transformed image is named normalized iris image. Then, Gabor filters with different parameters are applied and each filter response is encoded into one bit. More precisely, the filtering and encoding processes can be mathematically described by

$$\begin{aligned} b_i &= 1 & \text{if } \int_{\rho} \int_{\varphi} I_0(\rho, \varphi) g_i(\rho, \varphi) d\rho d\varphi \geq 0 \\ b_i &= 0 & \text{if } \int_{\rho} \int_{\varphi} I_0(\rho, \varphi) g_i(\rho, \varphi) d\rho d\varphi < 0 \end{aligned} \quad (5.1)$$

where I_0 is a normalized iris image, g_i is either a real or an imaginary part of a Gabor filter at a particular spatial location in the polar coordinate, b_i is a phase bit in the template and $i \in 1, 2, \dots, 2048$. Note that in addition to the spatial locations, not all g_i share the same parameters, e.g., spatial frequency, which is different from the filtering process employed in Venugopalan and Savvides's study [23]. For each of the phase bits, IrisCode calculates one mask bit which indicates whether the corresponding phase bit is corrupted by noise, such as eyelids and eyelashes, or not. Let m_i be the mask bit of b_i . $m_i = 1$ means that b_i is not corrupted and can be used in matching; otherwise b_i is corrupted and cannot be used in matching.

¹The work in this chapter has been published as "From the Perspective of CNN to Adversarial Iris Images." In 2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS), 2018.

Note that once $m_i = 0$, the original value of b_i is not necessary to be stored. In other words, for all $m_i = 0$, IrisCode can set either all corresponding b_i to zero or one without affecting the matching results. The previous studies did not take this point into consideration [23, 103, 107]. To match two given templates, p and q , IrisCode uses the normalized hamming distance

$$h(p, q) = \frac{\sum_{i=1}^{2048} (b_{pi} \oplus b_{qi}) \cap (m_{pi} \cap m_{qi})}{\sum_{i=1}^{2048} (m_{pi} \cap m_{qi})} \quad (5.2)$$

where $b_{pi}(b_{qi})$ and $m_{pi}(m_{qi})$ are respectively the i^{th} phase bit and the i^{th} mask bit of $p(q)$ and $\oplus$ and $\cap$ are respectively the bitwise XOR and bitwise AND operators for high speed matching. The summations in Eq. 5.2 are implemented using a coding table. For rotational invariant matching, one of the templates is shifted. More computational details about IrisCode can be found in [88–90]. In this chapter, the two-dimensional image and Gabor filters in Eq. 5.1 are expressed in discrete form as these matrices are expressed as vectors after lexicographic ordering. Therefore, all filtering operations are expressed as inner products. For the sake of convenience, we denote $I(\rho, \varphi) = I_0(\rho, \varphi)\rho$ and the discrete implementation of Eq. 5.1 can be written as

$$\begin{aligned} b_i &= 1 \quad \text{if} \quad g_i^T \leq 0 \\ b_i &= 0 \quad \text{if} \quad g_i^T < 0 \end{aligned} \quad (5.3)$$

Note that I and g_i are column vectors and T represents the transpose operator. In addition, the phase bits and the mask bits are also denoted respectively as vectors, $B = [b_1, b_2, \dots, b_{2048}]^T$ and $M = [m_1, m_2, \dots, m_{2048}]^T$.

FIGURE 5.1: The computational process of IrisCode.

FIGURE 5.2: Illustration of the filtering and coding processes in IrisCode as a convolutional neural network.

5.3 From a View of Convolutional Neural Network to an Unconstrained Formulation

The filtering and coding processes of IrisCode can be viewed as a convolutional neural network (CNN) with the Gabor filters in the first layer and the step function in the second layer illustrated in Figure 5.2. This network is non-differentiable and the gradient of the step function is zero everywhere, except for one point, for which gradient is undefined. Thus, the approaches generating adversarial examples for deep networks are not applicable. Using the logistic function

$$s(x) = \frac{1}{1 + e^{-kx}} \quad (5.4)$$

to approximate the step function, the network becomes differentiable. In Eq. 5.4, k is a parameter with a positive value controlling the shape of the function. The

logistic function has non-zero gradient everywhere. Thus, the generation problem can be formulated as:

$$I = \underset{X}{\operatorname{argmin}} \sum_{i=1}^{2048} (b_i - s(g_i^T X))^2 \quad (5.5)$$

where b_i is the i^{th} phase bit in a given template from IrisCode. X is the adversarial iris image we want to generate. Eq. 5.5 is an unconstrained minimization problem. Note that the squared L_2 norm is the same as hamming distance if two input vectors are binary. In Eq. 5.5, the squared L_2 norm is an approximation of the hamming distance used in IriCode because $S(g_i^T X)$ is not binary. If $|g_i^T X|$ is large enough and

$$\operatorname{sign}(g_i^T X) = 2(b_i - 0.5) \quad (5.6)$$

where $\operatorname{sign}(\cdot)$ is a sign function, $\underset{x}{\operatorname{argmin}} \sum_{i=1}^{2048} (b_i - S(g_i^T X))^2$ is approximately zero. Let $\hat{b}_i = 2(b_i - 0.5)$. Because X is a variable and its dimension is significantly higher than the number of filters, 2048, there exists some X such that $R = |g_i^T X|, \forall g_i$. Thus, Eq. 5.6 can be rewritten as

$$g_i^T X = R \hat{b}_i \quad (5.7)$$

Rewriting Eq. 5.7 in a matrix form, we obtain

$$\Psi X = R \hat{B} \quad (5.8)$$

where $\Psi = [g_1, g_2, \dots, g_{2048}]^T$ and $\hat{B} = [\hat{b}_1, \hat{b}_2, \dots, \hat{b}_{2048}]^T$. Eq. 5.8 is an underdetermined system of linear equations with many solutions. One of them can be

obtained by the pseudo inverse Ψ (Eq. 5.8) and the convex polyhedral cone formulation proposed by Kong [103] are in fact the same though they are derived from different points of view, one from a convolutional neural network and the other from geometry. As mentioned before, the convex polyhedral cone formulation does not offer high quality iris images.

5.4 From Unconstrained Minimization to Constrained Minimization

Though the formulation in Eq. 5.5 does not offer high quality iris images, it provides a starting point to modify the objective function to obtain better solutions. One of such modifications is to force X to be a real iris image. Let X_k be a real normalized iris image and $w(\cdot, \cdot)$ be a real-valued function which measures the dissimilarity between X_k and X . Eq. 5.5 can be modified to

$$I = \underset{x}{\operatorname{argmin}} \sum_{i=1}^{2048} (b_i - s(g_i X))^2 + \alpha w(X, X_k) \quad (5.9)$$

where α is a weight indicating the relative importance between the two terms. The logistic function in the first term is an approximation to the step function such that the objective function in Eq. 5.5 is differentiable and has non-zero gradient. Using the step function directly, the first term can be considered as constraints and the objective function in Eq. 5.9 can be rewritten as

$$\begin{aligned} I &= \underset{X}{\operatorname{argmin}} w(X, X_\kappa) \\ \text{s.t. } \quad &\hat{b}_i g_i^T > 0. \quad \forall i \end{aligned} \quad (5.10)$$

Note that these constraints are linear and $\hat{b}_i g_i^T X > 0$ implies $\text{sign}(g_i^T X) = 2(b_i - 0.5)$, except for $g_i^T X = 0$. Eq. 5.10 does not accept X satisfying $b_i g_i^T X = 0$ for any i as a solution. In fact, valid phase bits, whose corresponding $m_i = 1$, should not come from $g_i^T X = 0$, because they would be removed by the ordinal statistical threshold based on signal-to-noise ratio used in IrisCode. If w is convex, the constrained minimization formulation in Eq. 5.10 is also convex. In the experiments, we set w as a squared L_2 norm. Eq. 5.10 can be further rewritten as:

$$\begin{aligned} I = \underset{X}{\operatorname{argmin}} \quad & \| (X - X_\kappa) \|_2^2 \\ \text{s.t.} \quad & \hat{b}_i g_i^T X > 0. \quad \forall i \end{aligned} \quad (5.11)$$

and can be solved by quadratic programming solvers. This formulation does not use any information in the mask bits, M and assumes that the template stores all the original values of the phase bits, including those corrupted by noise, e.g., eyelashes. Furthermore, it does not provide any criterion to select X_κ . To take these points into consideration, the following constrained optimization formulation is finally proposed,

$$\begin{aligned} I = \underset{X}{\operatorname{argmin}} \quad & \| (X - X_k) \|_2^2 \\ \text{s.t.} \quad & \hat{b}_i g_i^T X > 0. \quad i \in c | m_c = 1 \end{aligned} \quad (5.12)$$

where X_κ is obtained from,

$$\kappa = \underset{j}{\operatorname{argmin}} \| M - M_j \|_1 \quad (5.13)$$

M_j is the mask bits of an iris image X_j in a given database for searching X_κ . M is the mask bits of the input template and m_c is the c^{th} bit of M . $\| \cdot \|_1$ denotes L_1

norm. $m_c = 1$ means that the corresponding phase bit is uncorrupted. Eq. 5.13 is used to search an iris image in a given database, whose eyelashes and eyelashes are roughly at the same locations as those in the target image. In the rest of the chapter, κ is called reference image and the proposed algorithm is called Convex Adversarial Iris Generation Algorithm (CAIG).

In the implementation, the quadratic programming solver in Matlab is used to solve Eq. 5.12. The solver provides an option to allow violate some of the constraints to obtain a solution closer to X_κ . In the experiments, the dimension of X is 32,768(512×64 pixels) and the average number of constraints in Eq. 5.12 is 1349. It is a large quadratic programming problem. Since IrisCode uses hamming distance to perform matching, it is not necessary to force all the phase bits in B and the phase bits derived from the generated iris to be the same. Thus, the option is taken up. More clearly, the parameter ConstraintTolerance in the quadprog function is set to default value of 10^{-8} .

FIGURE 5.3: Images without iris in the UBIRIS.v1 database.

FIGURE 5.4: Low quality samples used in the evaluation.

5.5 Experimental Results

Two iris databases, the UBIRIS.v1 and West Virginia University (WVU) iris databases [116, 117] were used to evaluate the proposed algorithm. The UBIRIS.v1

iris images were taken in a visible lighting environment, while the WVU iris images were taken in a near infrared lighting environment. The original images in the UBIRIS.v1 database are color images. Their red components were used in the experiments, because the wavelength of the red light is close to that of the near infrared and the iris texture in these components is clearest. The UBIRIS.v1 database stores 1,877 images from 241 irises, and the WVU iris database stores 3,099 iris images from 472 irises. All images in the WVU iris database were used in the experiments. As with the previous studies, 48 images from the UBIRIS.v1 database were automatically removed, because of their poor quality [103]. Some of them do not even have irises. Figure 5.3 shows some of these examples. Though some extremely low quality images were removed, many challenging samples were retained for evaluation. Figure 5.4 shows some of these samples.

To evaluate the proposed algorithm and compare it with the state-of-the-art methods, six reference and non-reference image quality indexes were utilized to examine the quality of the generated images and matching experiments were performed to check the bit errors due to the implementation. In addition, resultant images are also given for visual evaluation. The information hiding method (IH) [23], the graph-based method (GB) [107] and the convex polyhedral cone method (CPC) [103] were employed to compare with the proposed CAIG. In the experiments, all the original phase bits, including those corrupted by noise, are provided to GB and CPC methods such that they can be directly compared with the proposed CAIG. IH method was designed for binary iris templates which greatly differ from those of IrisCode. The differences were mentioned in Section 5.1. To apply it on templates of IrisCode for this evaluation, it has to be modified. The IH method requests iris template storing one bit for each pixel in the normalized iris image. Since there are only 2048 phase bits in templates of IrisCode and the size of the normalized iris image is 512×64 pixels, nearest neighbor interpolation was used to estimate other required bits. IH combines a reference image X_{IH} and a residual R_{IH} which

is derived from the input binary template to form its final result. More clearly, the IH method uses the equation $I_{IH} = (1 - \beta)R_{RI} + \beta X_{IH}$, where β is a parameter balancing X_{IH} and R_{RI} and I_{IH} is the final result. To make I_{IH} closer to a real iris image, β should be large. However, to retain more information from the template for matching, β should be small. Thus, in the experiments, $\beta = 0.3$ and $\beta = 0.7$ were examined. For the sake of convenience, IH(0.3) and IH(0.7) are used to denote IH using $\beta = 0.3$ and $\beta = 0.7$, respectively. Note that Venugopalan and Savvides also used $\beta = 0.7$ in their experiments. For fair comparison, the proposed algorithm and IH used the same reference image i.e., $X_{IH} = X_{\kappa}$.

In the first experiment, the normalized hamming distance in Eq. 5.2 was used to measure the percentage of bit difference between the template from the original iris image and the template from the corresponding CAIG result. Figure 5.5 shows the cumulative distributions of the normalized bit errors from the two databases. These bit errors were due to the optional parameter, ConstraintTolerance, in the solver but not due to the original formulation. The experimental results show that it causes bit error rate of 2.66 % for the UBIRIS.v1 database and bit error rate of 4.31% for the WVU database on average. These errors are neglectable, comparing with the thresholds commonly used in commercial iris recognition systems. The cumulative distributions of the normalized bit errors of the IH method are also provided in Figure 5.5. The average bit error rates of IH(0.3) are respectively 25.73% and 26.6% for the UBIRIS.v1 and WVU databases and the average bit error rates of IH(0.7) are respectively 26.8 % and 27.97% for the UBIRIS.v1 and WVU databases. Their bit errors are significantly higher than those of CAIG. Since CPC and GB methods force all the phase bits from their resultant images and the corresponding original images being the same, their cumulative distributions of bit errors are not given.

To evaluate the impact of the bit errors, IrisCode is used to match the original iris images and the resultant images from IH(0.3), IH(0.7) and CAIG. The other two

FIGURE 5.5: The cumulative distributions of normalized bit error from the (A) UBIRIS.v1 database (B) WVU database.

methods, CPC and GB, do not have bit errors and therefore, they are not included in this experiment. In this matching experiment, the CAIG and IH resultant images were not allowed to match their parent images, whose phase and mask bits were inputted to CAIG and IH for generating the resultant images. Receiver operating characteristic (ROC) curves are used as a performance index. For the same database, all the ROC curves were computed from the same imposter distribution that determines the false acceptance rate of a system when a threshold is given. Figure 5.6 shows clearly that the bit errors have almost no effect on CAIG matching performance. However, the performance of IH(0.3) and IH(0.7) has significant drops compared with the original ROC curves.

To evaluate the visual quality, Figure 5.7 shows eight groups of normalized iris images from the two databases. In each group of the images, the first and the second images are respectively the original normalized iris image and the reference image, X_κ in Eqs. 5.12–5.13. The 3rd to 7th images are respectively results from CAIG, GB, CPC, IH(0.3) and IH(0.7). GB and CPC methods do not provide contrast and image DC for image display. To display them, the contrast and DC from the original images are used. CPC method does not provide any results similar to real iris images (the 5th image in each group). Its results are similar

FIGURE 5.6: The ROC curves of the information hiding method and the proposed algorithm for (A) UBIRIS.v1 database (B) WVU database.

to high frequency noise. The IH(0.3) and IH(0.7) results have many thick black curves. The IH(0.7) results (the 7th image in each group) are relatively better. Some eyelids, which are obtained from the reference images, can be seen. However, their visual quality is still very low. GB method provides higher visual quality results comparing with CPC, IH(0.3) and IH(0.7) methods. However, they are still very noisy with many high frequency patterns. The CAIG results are much better than others. They are very similar to the reference images. Some of them have no visible difference. The proposed CAIG algorithm outperforms all the other methods and generates high quality iris images from the templates of IrisCode.

To compare the proposed algorithm with the other methods objectively, three reference and three non-reference image quality indexes which are widely used for image quality assessment are selected for this experiment. The three reference image quality indexes are Mean-Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index Measure (SSIM). These metrics require two images from the same image computer—a reference image and a reconstructed image. MSE measures the average squared difference between two images. PSNR is derived from the mean square error and indicates the maximum pixel intensity

of the reconstructed image to the power of the distortion. MSE and PSNR metrics are simple to calculate but might not align well with the human perception of quality. SSIM combines structure, luminance, and contrast into a quality score to measure the perceptive difference between two images. In this metric, structures are patterns of pixel intensities among neighboring pixels. The three non-reference image quality indexes that only take a reconstructed image as input are, Distortion Identification-based Image Verity and INtegrity Evaluation (DIIVINE), Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE), Naturalness Image Quality Evaluator (NIQE) [118–122]. NIQE makes use of measurable deviations from statistical regularities observed in natural images, without training on human-rated distorted images. BRISQUE uses scene statistics of locally normalized luminance

(A)

(B)

FIGURE 5.7: Visual comparison of different results. (A) UBIRIS.v1 database and (B) WVU database. There are seven images in each group. The first and the second images are respectively the original normalized iris image and the reference image, X_{κ} in Eqs. 5.12–5.13. The 3rd to 7th images are respectively CAIG, GB, CPC, IH(0.3) and IH(0.7) results.

coefficients to quantify possible losses of naturalness in the image due to the presence of distortions. DIIVINE utilizes a 2-stage framework that first identifies the distortion afflicting the image and then performs distortion-specific quality assessment. The results from the three reference image quality indexes are given in Table 5.1. Each value in the table is the mean of an index from one database. The CPC and GP resultant images are computed from only the original images but not the reference images. Thus, the reference image quality indexes are used to measure the visual similarity between their resultant images and the original images. The

FIGURE 5.8: The distributions of non-reference image quality indexes. The first and the second column are respectively distributions of the UBIRIS.v1 and WVU databases. The 1st-3rd rows are respectively distributions of BRISQUE, NIQE and DIIVINE.

TABLE 5.1: Comparisons of different methods using reference quality image indexes. Each value in the table is the mean of an index for one database.

UBIRIS.v1		SSIM	MSE	PSNR
CAIG	original image	0.1667	1629.9	17.9293
	reference image	0.9085	9.9	38.7925
GB	original image	0.1378	503.9	21.9022
CPC	original image	0.016	2111.4	15.8563
IH(0.7)	original image	0.118	1891.1	17.2634
	reference image	0.3841	728.7	20.2383
IH(0.3)	original image	0.087	2010	16.8302
	reference image	0.1125	954.7	19.1574
WVU		SSIM	MSE	PSNR
CAIG	original image	0.161	2623.6	14.9756
	reference image	0.8618	31.6	33.995
GB	original image	0.1127	1528.6	17.3192
CPC	original image	0.0177	4823.3	12.3956
IH(0.7)	original image	0.1007	3095.2	14.4757
	reference image	0.3437	1726.8	16.5721
IH(0.3)	original image	0.0869	3279.4	14.178
	reference image	0.1068	2260.5	15.5673

IH(0.3), IH(0.7) and CAIG results are based on information in the original images and the reference images. Thus, the indexes are used to measure the visual similarity between their resultant images and the original images and also between their resultant images and the reference images. Table 5.1 indicates that none of the resultant images are similar to the original images. For the original images, all their PSNR and SSIM are very low and their MSE are high. For the reference images, IH(0.7) obtains slightly better results. However, their similarity is still very low. Table 5.1 shows clearly that the results generated by the proposed algorithm and the reference images are very similar, which aligns with our visual evaluation in Figure 5.7.

The IH(0.3), IH(0.7) and CAIG results are in fact generated from two images, the original and the reference images. Using the reference image quality indexes to measure their quality may not be ideal. Thus, three non-reference image quality indexes, DIIVINE, BRISQUE and NIQE are also employed. The results are listed

TABLE 5.2: Comparisons of different methods using non-reference quality image indexes. Each value in the table is the mean of an index for one database.

UBIRIS.v1	DIIVINE	BRISQUE	NIQE
CAIG	33.276	17.5265	5.3778
GB	39.4657	29.4612	10.4407
CPC	38.6627	43.2663	19.864
IH(0.7)	53.6834	45.6109	9.6396
IH(0.3)	40.1617	32.3897	6.0627
Real iris images	34.3159	18.7008	5.2374
WVU	DIIVINE	BRISQUE	NIQE
CAIG	27.1198	23.5846	5.0566
GB	48.6509	32.1407	10.9212
CPC	44.8651	43.2966	19.9975
IH(0.7)	50.8242	46.0282	8.5813
IH(0.3)	32.519	39.3584	5.6957
Real iris images	28.9947	27.5465	5.2685

in Table 5.2. In addition to the resultant images of the methods, the quality values from real iris images are also provided for comparison. Lower index values represent higher quality. Clearly, the proposed CAIG algorithm outperforms the other methods. The quality indexes of the CAIG results and the real iris images are very close. Figure 5.8 shows the distributions of the non-reference image quality indexes. The distributions of the real iris images and the CAIG results are highly overlapping, meaning that their quality is very similar to real iris images.

5.6 Summary

IrisCode, the most impactful iris recognition algorithm, has been extensively deployed to large-scale personal identification programmes. Each aspect related to IrisCode, in particularly security threats, has to be studied thoroughly. The previous studies demonstrated that low quality iris images can be generated from the templates of IrisCode. It was not clear whether it is possible to produce high

quality iris images from them. In this chapter, the feature extraction and encoding processes in IrisCode are viewed as a non-differentiable shallow convolutional neural network. From this perspective, an unconstrained optimization formulation is developed. Its solution is same as that of the convex polyhedral cone method proposed before. Based on this formulation, a convex constrained optimization formulation, which generates adversarial iris images, is derived. Experimental results on two public databases demonstrate that high quality iris images can be generated and the proposed CAIG algorithm significantly outperforms the previous methods.

Chapter 6

Conclusion

Though numerous works have been done on identifying new threats against various DNNs models, there are still some popular and widely utilized architectures that have not been fully studied. This thesis focuses on generating adversarial examples against three different architectures to analyze their security risks under new types of adversarial attacks. In this chapter, we conclude the main contributions of this thesis and suggest directions for future work.

6.1 Major Contributions

Our main contributions can be stated as follows:

- Object detection: we propose two new algorithms to generate adversarial examples that can successfully attack an object detector digitally and physically. To the best of our knowledge, they are the first two methods, which can attack object detectors without modifying the target object. The first algorithm generates an adversarial border, which is placed around the target object to fool object detectors. The generated border makes the target object

cannot be detected by object detectors without changing it, while the adversarial border makes the target object unnatural to human eyes. To make the attack looks more natural, a new type of adversarial example named adversarial signboard, which looks like a signboard, is generated by the second algorithm. The adversarial signboard is placed beneath a target object to mislead object detectors. The experimental results show that these two types of adversarial examples can effectively attack object detectors in both black-box and white-box scenarios.

- Non-local blocks: we introduced two new attacks named DA and AA against object detectors with a non-local block, which is the first study of analyzing risks from non-local blocks. DA aims at fooling an object detector with a non-local block such that it is unable to detect a target object category while AA aims at misleading the object detector into detecting a predefined object category, not in images. In contrast to other attacks against object detectors, DA and AA can be implemented at a relatively long distance. The experimental results show that DA and AA can attack Faster R-CNN both digitally and physically with a high probability. To explain the new threats caused by non-local blocks, we provide empirical and theoretical studies on the reception field of CNN models with and without non-local blocks.
- Iris recognition: by expanding the definition of adversarial example in deep learning to iris recognition, we formulated the problem of generating iris images from templates as a constrained convex optimization problem. For a fixed template, the formulation utilizes an input image selected by a search function as an information carrier. The experimental results demonstrate that the proposed algorithm can generate high-quality iris images, which can be matched with real iris images from target eyes. The generated iris images significantly outperform the iris images generated by previous methods in terms of visual quality and six image quality metrics.

6.2 Future Work

Though adversarial examples have been studied extensively for several years, there are still many problems needed to be investigated.

On the attacking side, we summarize three directions that should be further studied. The first direction is based on architecture. Current studies of adversarial examples mainly focus on image classification. Studies on other architectures are still insufficient. Therefore, the first direction of our future work is to identify the potential threats of architectures that have not been examined, especially those newly proposed architectures. The second direction is based on the distance metric. We noticed that most of the proposed attacks utilize $L_p(p = 1, 2, \text{inf})$ as the distance metric which is not directly related to visual perception. According to the definition of adversarial examples, a better distance metric can be used to construct stronger adversarial examples. Therefore, the second direction of our future work is to generate stronger adversarial examples using different distance metrics. The third direction of our work is based on assumptions. We noticed that most of the current attacks are under the white-box assumption. Only a small number of studies focus on black-box attacks. Their strategies are either relying on the transferability of adversarial examples or generating the adversarial examples based on the output. However, adversarial examples are not always transferable and the attacker may only have limited information about the output. For example, the attacker may only have the output label of a classifier instead of the probability of each class. The above-mentioned problems have not been fully resolved. Therefore, the third direction of our future work is to propose stronger black-box attacks, which can transfer to various models and be performed under extremely limited knowledge.

On the defense side, the first potential direction is to improve the security of object detectors against the proposed attacks. It can be done by designing a sub-network

to automatically detecting the adversarial examples or to remove the adversarial noise in an image. Though lots of countermeasures have been proposed, almost all the defense methods are effective only for part of the attacks. They are not very effective for unseen attacks. At the same time, with more and more attacks being devised in other areas, new defenses for these areas are urgently required. Thus, strong defense algorithms, which are effective for various attack methods and even for different applications could be another research direction. We hope that through studying attacks and defenses of adversarial examples, we can obtain some hints to understand the existence and transferability of adversarial examples.

List of Author's Publications

Conference Proceedings

- **Huang, Yi**, Wang, Fan, Adams Wai-Kin Kong, and Kwok-Yan Lam. “New Threats against Object Detector with Non-Local Block”, ECCV, 2020
- **Huang, Yi**, Adams Wai-Kin Kong, and Kwok-Yan Lam. “Adversarial Signboard against Object Detector.” In BMVC, 2019.
- **Huang, Yi**, Adams Wai-Kin Kong, and Kwok-Yan Lam. “Attacking object detectors without changing the target object.” In Pacific Rim International Conference on Artificial Intelligence, pp. 3-15. Springer, Cham, 2019.
- **Huang, Yi**, Adams Wai Kin Kong, and Kwok-Yan Lam. “From the Perspective of CNN to Adversarial Iris Images.” In 2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS), pp. 1-10. IEEE, 2018.

Bibliography

- [1] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. URL <http://arxiv.org/abs/1412.6572>. [xi](#), [13](#), [14](#)
- [2] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773, 2017. [xi](#), [13](#), [17](#), [18](#)
- [3] Zhengli Zhao, Dheeru Dua, and Sameer Singh. Generating natural adversarial examples. In *International Conference on Learning Representations*, 2018. URL <https://openreview.net/forum?id=H1BLjgZCb>. [xi](#), [13](#), [18](#), [19](#)
- [4] Jiajun Lu, Hussein Sibai, Evan Fabry, and David A. Forsyth. No need to worry about adversarial examples in object detection in autonomous vehicles. *CoRR*, abs/1707.03501, 2017. [xi](#), [23](#), [24](#), [30](#)
- [5] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan L. Yuille. Adversarial examples for semantic segmentation and object detection. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1378–1387, 2017. [xi](#), [23](#), [25](#), [50](#)
- [6] Jiajun Lu, Hussein Sibai, and Evan Fabry. Adversarial examples that fool detectors. *CoRR*, abs/1712.02494, 2017. [xi](#), [24](#), [25](#), [26](#), [50](#), [51](#), [55](#)
- [7] Shang-Tse Chen, Cory Cornelius, Jason Martin, and Duen Horng Chau. Shapeshifter: Robust physical adversarial attack on faster r-cnn object detector. In *ECML/PKDD*, 2018. [xi](#), [25](#), [26](#), [50](#), [51](#), [55](#)
- [8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. [1](#)
- [9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. [1](#)
- [10] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen,

- Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97, 2012. 1
- [11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. 2017. URL <https://arxiv.org/pdf/1706.03762.pdf>. 1
- [12] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014. 1
- [13] The history of amazon’s recommendation algorithm. <https://www.amazon.science/the-history-of-amazons-recommendation-algorithm>. 1
- [14] How google search (probably) uses machine learning. <https://www.embedded-computing.com/guest-blogs/how-google-search-probably-uses-machine-learning>. 1
- [15] Paypal feeds the dl beast with huge vault of fraud data. <https://www.datanami.com/2019/06/03/paypal-feeds-the-dl-beast-with-huge-vault-of-fraud-data/>. 1
- [16] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 2, 7, 10, 12, 13
- [17] Nilesch Dalvi, Pedro Domingos, Sumit Sanghai, and Deepak Verma. Adversarial classification. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 99–108, 2004. 8, 9
- [18] Daniel Lowd and Christopher Meek. Adversarial learning. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 641–647, 2005. 8
- [19] Daniel Lowd and Christopher Meek. Good word attacks on statistical spam filters. In *CEAS*, volume 2005, 2005. 8
- [20] Blaine Nelson, Benjamin IP Rubinstein, Ling Huang, Anthony D Joseph, Steven J Lee, Satish Rao, and JD Tygar. Query strategies for evading convex-inducing classifiers. *The Journal of Machine Learning Research*, 13(1):1293–1332, 2012. 8
- [21] Battista Biggio, Iginio Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio 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. 8

- [22] Pavel Laskov et al. Practical evasion of a learning-based classifier: A case study. In *2014 IEEE symposium on security and privacy*, pages 197–211. IEEE, 2014. 8
- [23] Shreyas Venugopalan and Marios Savvides. How to generate spoofed irises from an iris code template. *IEEE Transactions on Information Forensics and Security*, 6(2):385–395, 2011. 8, 74, 76, 77, 83
- [24] C. middlehurst, china unveils world’s first facial recognition atm. <http://www.telegraph.co.uk/news/worldnews/asia/china/11643314/China-unveils-worlds-first-facial-recognition-ATM.html>, 2015. 9
- [25] E. ackerman, how drive.ai is mastering autonomous driving with deep learning. <https://spectrum.ieee.org/cars-that-think/transportation/self-driving/>. 9
- [26] Battista Biggio and Fabio Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317–331, 2018. 10
- [27] Marco Barreno, Blaine Nelson, Anthony D Joseph, and J Doug Tygar. The security of machine learning. *Machine Learning*, 81(2):121–148, 2010. 10
- [28] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519, 2017. 11, 13, 20
- [29] Andras Rozsa, Ethan M Rudd, and Terrance E Boult. Adversarial diversity and hard positive generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 25–32, 2016. 13, 14
- [30] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017. 13, 14
- [31] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016. 13, 14
- [32] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018. 13, 15
- [33] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroSecP)*, pages 372–387. IEEE, 2016. 13, 15, 16

- [34] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016. [13](#), [16](#)
- [35] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017. [13](#), [17](#), [21](#)
- [36] Shumeet Baluja and Ian Fischer. Learning to attack: Adversarial transformation networks. In *Proceedings of AAAI-2018*, 2018. URL <http://www.esprockets.com/papers/aaai2018.pdf>. [13](#), [18](#)
- [37] Omid Poursaeed, Isay Katsman, Bicheng Gao, and Serge Belongie. Generative adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4422–4431, 2018. [13](#), [19](#)
- [38] Yang Song, Rui Shu, Nate Kushman, and Stefano Ermon. Constructing unrestricted adversarial examples with generative models. In *Advances in Neural Information Processing Systems*, pages 8312–8323, 2018. [13](#), [19](#)
- [39] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL <https://openreview.net/forum?id=Sys6GJqxl>. [13](#), [20](#)
- [40] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 15–26, 2017. [13](#), [21](#)
- [41] Jonathan Uesato, Brendan O’Donoghue, Pushmeet Kohli, and Aaron van den Oord. Adversarial risk and the dangers of evaluating against weak attacks. volume 80 of *Proceedings of Machine Learning Research*, pages 5025–5034, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL <http://proceedings.mlr.press/v80/uesato18a.html>. [13](#), [21](#)
- [42] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. [14](#), [17](#)
- [43] Alexey Kurakin, Ian Goodfellow, Samy Bengio, Yinpeng Dong, Fangzhou Liao, Ming Liang, Tianyu Pang, Jun Zhu, Xiaolin Hu, Cihang Xie, et al. Adversarial attacks and defences competition. In *The NIPS’17 Competition: Building Intelligent Systems*, pages 195–231. Springer, 2018. [15](#)

- [44] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram 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. [17](#)
- [45] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678, 2014. [17](#)
- [46] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. [17](#), [49](#)
- [47] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. [17](#)
- [48] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML’17*, page 214–223. JMLR.org, 2017. [18](#)
- [49] James C Spall et al. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE transactions on automatic control*, 37(3):332–341, 1992. [21](#)
- [50] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. Defense against adversarial attacks using high-level representation guided denoiser. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1778–1787, 2018. [21](#)
- [51] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan L. Yuille. Mitigating adversarial effects through randomization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL <https://openreview.net/forum?id=Sk9yuql0Z>.
- [52] Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. In *International Conference on Learning Representations*, 2018. [21](#)
- [53] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali 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. [22](#)
- [54] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. [22](#), [23](#)

- [55] Pedro Tabacof, Julia Tavares, and Eduardo Valle. Adversarial images for variational autoencoders. *arXiv preprint arXiv:1612.00155*, 2016. 26
- [56] Jernej Kos, Ian Fischer, and Dawn Song. Adversarial examples for generative models. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 36–42. IEEE, 2018. 26
- [57] Nicolas Papernot, Patrick McDaniel, Ananthram Swami, and Richard Harang. Crafting adversarial input sequences for recurrent neural networks. In *MILCOM 2016-2016 IEEE Military Communications Conference*, pages 49–54. IEEE, 2016. 27
- [58] Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel. Adversarial attacks on neural network policies. *arXiv preprint arXiv:1702.02284*, 2017. 27
- [59] Yen-Chen Lin, Zhang-Wei Hong, Yuan-Hong Liao, Meng-Li Shih, Ming-Yu Liu, and Min Sun. Tactics of adversarial attack on deep reinforcement learning agents. *arXiv preprint arXiv:1703.06748*, 2017. 27
- [60] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael 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. ACM, 2016. 27
- [61] Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. 2015. 27
- [62] Weilin Xu, Yanjun Qi, and David Evans. Automatically evading classifiers: A case study on pdf malware classifiers. 01 2016. doi: 10.14722/ndss.2016.23115. 27
- [63] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014. URL <http://arxiv.org/abs/1312.6199>. 28, 50
- [64] Nicolas Papernot, Patrick D. McDaniel, and Ian J. Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *CoRR*, abs/1605.07277, 2016. 28
- [65] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018. 34
- [66] Authors. Attacking object detectors without changing the target object, 2019. IJCAI-19 submission ID 2455. Supplied as additional material IJCAI-19.pdf. 42

- [67] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *Advances in neural information processing systems*, pages 4898–4906, 2016. [49](#), [64](#), [65](#), [66](#)
- [68] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 60–65. IEEE, 2005. [49](#)
- [69] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018. [49](#), [50](#), [52](#), [56](#)
- [70] Zhengzheng Tu, Yan Ma, Chenglong Li, Jin Tang, and Bin Luo. Edge-guided non-local fully convolutional network for salient object detection. *arXiv preprint arXiv:1908.02460*, 2019. [50](#)
- [71] Xin Xu and Jie Wang. Extended non-local feature for visual saliency detection in low contrast images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018.
- [72] Hila Levi and Shimon Ullman. Efficient coarse-to-fine non-local module for the detection of small objects. *arXiv preprint arXiv:1811.12152*, 2018.
- [73] Mohammad Shokri, Ahad Harati, and Kimya Taba. Salient object detection in video using deep non-local neural networks. *arXiv preprint arXiv:1810.07097*, 2018. [50](#)
- [74] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Non-local graph convolutional networks for skeleton-based action recognition. *arXiv preprint arXiv:1805.07694*, 2018. [50](#)
- [75] Guyue Hu, Bo Cui, and Shan Yu. Skeleton-based action recognition with synchronous local and non-local spatio-temporal learning and frequency attention. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1216–1221. [50](#)
- [76] Xingyu Liao, Lingxiao He, Zhouwang Yang, and Chi Zhang. Video-based person re-identification via 3d convolutional networks and non-local attention. In *Asian Conference on Computer Vision*, pages 620–634. Springer, 2018. [50](#)
- [77] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019. [50](#)

- [78] Kaiyu Yue, Ming Sun, Yuchen Yuan, Feng Zhou, Errui Ding, and Fuxin Xu. Compact generalized non-local network. In *Advances in Neural Information Processing Systems*, pages 6510–6519, 2018. [50](#)
- [79] Yongyi Tang, Xing Zhang, Lin Ma, Jingwen Wang, Shaoxiang Chen, and Yu-Gang Jiang. Non-local netvlad encoding for video classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018. [50](#)
- [80] Canmiao Fu, Wenjie Pei, Qiong Cao, Chaopeng Zhang, Yong Zhao, Xiaoyong Shen, and Yu-Wing Tai. Non-local recurrent neural memory for supervised sequence modeling. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6311–6320, 2019. [50](#)
- [81] Lu Chi, Guiyu Tian, Yadong Mu, Lingxi Xie, and Qi Tian. Fast non-local neural networks with spectral residual learning. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 2142–2151, 2019. [50](#)
- [82] Lei Chen, Hong Song, Qiang Li, Yutao Cui, Jian Yang, and Xiaohua Tony Hu. Liver segmentation in ct images using a non-local fully convolutional neural network. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 639–642. IEEE, 2019. [50](#)
- [83] Michał Zajac, Konrad Zolna, Negar Rostamzadeh, and Pedro O Pinheiro. Adversarial framing for image and video classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 10077–10078, 2019. [50](#)
- [84] Yi Huang, Adams Wai-Kin Kong, and Kwok-Yan Lam. Attacking object detectors without changing the target object. In *Pacific Rim International Conference on Artificial Intelligence*, pages 3–15. Springer, 2019. [50](#), [51](#), [55](#)
- [85] Yi Huang, Adams Wai Kin Kong, and Kwok-Yan Lam. Adversarial sign-board against object detector. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2019. [50](#), [51](#)
- [86] Dawn Song, Kevin Eykholt, Ivan Evtimov, Earlene Fernandes, Bo Li, Amir Rahmati, Florian Tramèr, Atul Prakash, and Tadayoshi Kohno. Physical adversarial examples for object detectors. In *12th {USENIX} Workshop on Offensive Technologies ({WOOT} 18)*, 2018. [50](#), [51](#), [55](#)
- [87] Xinlei Chen and Abhinav Gupta. An implementation of faster rcnn with study for region sampling. *arXiv preprint arXiv:1702.02138*, 2017. [55](#)
- [88] John G Daugman. High confidence visual recognition of persons by a test of statistical independence. *IEEE transactions on pattern analysis and machine intelligence*, 15(11):1148–1161, 1993. [73](#), [77](#)
- [89] John Daugman. How iris recognition works. In *The essential guide to image processing*, pages 715–739. Elsevier, 2009. [73](#), [74](#)

- [90] John Daugman. New methods in iris recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(5):1167–1175, 2007. 73, 77
- [91] John Daugman. 600 million citizens of india are now enrolled with biometric id. *SPIE newsroom*, 7, 2014. 73
- [92] Unique identification authority of india, government of india. <http://web.archive.org/web/20080207010024/http://www.808multimedia.com/winnt/kernel.htm>. 73
- [93] Major international deployments of the iris recognition algorithms: a billion persons. <http://www.cl.cam.ac.uk/~jgd1000/national-ID-deployments.htm>. 73
- [94] Wfp introduces iris scan technology to provide food assistance to syrian refugees in zaatari, world food programme. <https://www.wfp.org/news/news-release/wfp-introduces-innovative-iris-scan-technology-provide-food-assistance-syr>. 73
- [95] Lunke Fei, Bob Zhang, Yong Xu, and Liping Yan. Palmprint recognition using neighboring direction indicator. *IEEE transactions on human-machine systems*, 46(6):787–798, 2016. 74
- [96] Z Sun, T Tan, Y Wang, and SZ Li. Ordinal palmprint representation for personal identification. 2005. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- [97] Yang Hu, Konstantinos Sirlantzis, and Gareth Howells. Optimal generation of iris codes for iris recognition. *IEEE Transactions on Information Forensics and Security*, 12(1):157–171, 2016. 74
- [98] Nianfeng Liu, Jing Liu, Zhenan Sun, and Tieniu Tan. A code-level approach to heterogeneous iris recognition. *IEEE Transactions on Information Forensics and Security*, 12(10):2373–2386, 2017.
- [99] Hugo Proença. Iris recognition: What is beyond bit fragility? *IEEE Transactions on Information Forensics and Security*, 10(2):321–332, 2014.
- [100] Xiaoming Xi, Lu Yang, and Yilong Yin. Learning discriminative binary codes for finger vein recognition. *Pattern Recognition*, 66:26–33, 2017.
- [101] Wenming Yang, Xiaola Huang, Fei Zhou, and Qingmin Liao. Comparative competitive coding for personal identification by using finger vein and finger dorsal texture fusion. *Information sciences*, 268:20–32, 2014.
- [102] Karen P Hollingsworth, Kevin W Bowyer, and Patrick J Flynn. Improved iris recognition through fusion of hamming distance and fragile bit distance. *IEEE transactions on pattern analysis and machine intelligence*, 33(12):2465–2476, 2011. 74

- [103] Adams Wai-Kin Kong. Modeling iricode and its variants as convex polyhedral cones and its security implications. *IEEE transactions on image processing*, 22(3):1148–1160, 2012. [75](#), [77](#), [80](#), [83](#)
- [104] John Daugman. Information theory and the iricode. *IEEE transactions on information forensics and security*, 11(2):400–409, 2015.
- [105] Adams Wai-Kin Kong. A statistical analysis of iricode and its security implications. *IEEE transactions on pattern analysis and machine intelligence*, 37(3):513–528, 2014. [74](#)
- [106] Karen P Hollingsworth, Kevin W Bowyer, and Patrick J Flynn. The best bits in an iris code. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):964–973, 2008. [74](#)
- [107] Adams Wai Kin Kong. Iricode decompression based on the dependence between its bit pairs. *IEEE transactions on pattern analysis and machine intelligence*, 34(3):506–520, 2011. [75](#), [77](#), [83](#)
- [108] Adams WK Kong, David Zhang, and Mohamed S Kamel. An analysis of iricode. *IEEE transactions on image processing*, 19(2):522–532, 2009. [74](#)
- [109] Hugo Proença and Joao C Neves. Irina: Iris recognition (even) in inaccurately segmented data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 538–547, 2017. [74](#)
- [110] Javier Galbally, Arun Ross, Marta Gomez-Barrero, Julian Fierrez, and Javier Ortega-Garcia. Iris image reconstruction from binary templates: An efficient probabilistic approach based on genetic algorithms. *Computer Vision and Image Understanding*, 117(10):1512–1525, 2013. [74](#)
- [111] M Braithwaite, UC von Seelen, J Cambier, J Daugman, R Class, R Moore, and I Scott. Authentication using application-specific biometric templates. *WO2002/095657*, 2002. [74](#)
- [112] Emanuele Maiorana, Gabriel Emile Hine, and Patrizio Campisi. Hill-climbing attacks on multibiometrics recognition systems. *IEEE Transactions on Information Forensics and Security*, 10(5):900–915, 2014. [74](#)
- [113] Sheng Li and Alex C Kot. An improved scheme for full fingerprint reconstruction. *IEEE Transactions on Information Forensics and Security*, 7(6):1906–1912, 2012. [75](#)
- [114] Kai Cao and Anil K Jain. Learning fingerprint reconstruction: From minutiae to image. *IEEE Transactions on information forensics and security*, 10(1):104–117, 2014. [75](#)
- [115] Guangcan Mai, Kai Cao, Pong C Yuen, and Anil K Jain. Face image reconstruction from deep templates. *arXiv preprint arXiv:1703.00832*, 2017. [75](#)

- [116] Hugo Proença and Luís A Alexandre. Ubiris: A noisy iris image database. In *International Conference on Image Analysis and Processing*, pages 970–977. Springer, 2005. [75](#), [82](#)
- [117] Arun Ross and Samir Shah. Segmenting non-ideal irises using geodesic active contours. In *2006 Biometrics Symposium: Special Session on Research at the Biometric Consortium Conference*, pages 1–6. IEEE, 2006. [82](#)
- [118] Anush Krishna Moorthy and Alan Conrad Bovik. Blind image quality assessment: From natural scene statistics to perceptual quality. *IEEE transactions on Image Processing*, 20(12):3350–3364, 2011. [87](#)
- [119] K. moorthy and a. c. bovik, divine software release. http://live.ece.utexas.edu/research/quality/DIIVINE_release.zip, 2010.
- [120] Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on image processing*, 21(12):4695–4708, 2012.
- [121] Anish Mittal, Anush K Moorthy, and Alan C Bovik. Blind/referenceless image spatial quality evaluator. In *2011 conference record of the forty fifth asilomar conference on signals, systems and computers (ASILOMAR)*, pages 723–727. IEEE, 2011.
- [122] Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. Making a “completely blind” image quality analyzer. *IEEE Signal processing letters*, 20(3):209–212, 2012. [87](#)