ATTACK INDEPENDENT PERCEPTUAL IMPROVEMENT OF ADVERSARIAL EXAMPLES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS OF
THE MIDDLE EAST TECHNICAL UNIVERSITY
BY

BERAT TUNA KARLI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF INFORMATION SYSTEMS

DECEMBER 2022

Attack Independent Perceptual Improvement of Adversarial Examples

submitted by **BERAT TUNA KARLI** in partial fulfillment of the requirements for the degree of **Master of Science in Information Systems Department, Middle East Technical University** by,

Prof. Dr. Banu Günel Kılıç _____
Dean, **Graduate School of Informatics**

Prof. Dr. Altan Koçyiğit _____
Head of Department, **Information Systems**

Prof. Dr. Alptekin Temizel _____
Supervisor, **Modelling and Simulation, METU**

Examining Committee Members:

Assoc. Prof. Dr. Erhan Eren
Information Systems, METU _____

Prof. Dr. Alptekin Temizel
Modelling and Simulation, METU _____

Assist. Prof. Dr. Ayşegül Dündar
Computer Science, Bilkent University _____

Date: 23.12.2022

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Berat Tuna Karlı

Signature :

ABSTRACT

ATTACK INDEPENDENT PERCEPTUAL IMPROVEMENT OF ADVERSARIAL EXAMPLES

Karlı, Berat Tuna

M.S., Department of Information Systems

Supervisor: Prof. Dr. Alptekin Temizel

December 2022, 40 pages

Deep Neural networks (DNNs) are used in a variety of domains with great success, however, it has been proven that these networks are vulnerable to additive non-arbitrary perturbations. Regarding this fact, several attack and defense mechanisms have been developed; nevertheless, adding crafted perturbations has a negative effect on the perceptual quality of images. This study aims to improve the perceptual quality of adversarial examples independent of attack type and the integration of two attack agnostic techniques is proposed for this purpose. The primary technique, Normalized Variance Weighting, aims to improve the perceptual quality of adversarial attacks by applying a variance map to intensify the perturbations in the high-variance zones. This method could be applied to existing adversarial attacks without any additional overhead except a matrix multiplication. The secondary technique, the Minimization Method, minimizes the perceptual distance of the successful adversarial example to improve its perceptual quality. This technique could be applied to adversarial samples generated using any type of adversarial attack. Since the primary method is applied during the attack and the secondary method is applied after the attack, these two separate methods could be used together in an integrated adversarial attack setting. It is shown that adversarial examples generated from the integration of these methods exhibit the best perceptual quality measured in terms of the LPIPS metric.

Keywords: deep learning, adversarial attacks, perceptual quality, image classification

ÖZ

SALDIRI TİPİ BAĞIMSIZ ÇEKİŞMELİ ÖRNEKLERİN ALGISAL İYİLEŞTİRMESİ

Karlı, Berat Tuna

Yüksek Lisans, Bilişim Sistemleri Bölümü

Tez Yöneticisi: Prof. Dr. Alptekin Temizel

Aralık 2022, 40 sayfa

Günümüzde derin öğrenme birçok farklı alana uygulanmaktadır. Fakat, bu derin ağların zayıflıklarının olduğu ve bu zayıflıkların rastgele olmayan gürültüler kullanılarak çekişmeli örneklerle saldırıya uğratılabileceği kanıtlanmıştır. Bu süreçte birçok saldırı ve savunma yöntemleri geliştirilmiştir. Fakat, gürültü eklenilen çekişmeli örneklerin görüntü kalitesi azalmaktadır. Bu çalışma çekişmeli örneklerdeki algısal kaliteyi artırmayı hedeflemektedir ve bu amaçla saldırı tipi bağımsız çalışabilen iki ayrı yöntemin birleşimi önerilmiştir. Birincil yöntem (Normalleştirilmiş Varyans Ağırlıklandırma), yüksek varyans bölgelerindeki bozulmaları yoğunlaştırmak için varyans haritası uygulayarak çekişmeli saldırıların algısal kalitesini iyileştirmeyi amaçlar. Bu yöntem, matris çarpımı dışında hesapsal ek bir maliyet getirmeden herhangi bir saldırı tipine zahmetsizce uygulanabilir. İkincil yöntem (Minimizasyon Metodu), algısal kaliteyi iyileştirmek için başarılı çekişmeli örneğin algısal mesafesini en aza indirir. Bu teknik, herhangi bir tür çekişmeli saldırı ile üretilmiş olan çekişmeli örneklere uygulanabilir. Birincil yöntem saldırı sırasında ve ikincil yöntem saldırıdan sonra uygulandığı için bu iki ayrı yöntem aynı saldırı içinde tümleştirilebilir. Bu yöntemlerin birleşiminden elde edilen çekişmeli örneklerin algısal kalite ölçütlerine göre en iyi sonuçları ortaya koyduğu gösterilmiştir.

Anahtar Kelimeler: derin öğrenme, çekişmeli saldırılar, algısal kalite, imge sınıflandırma

Dedication here

ACKNOWLEDGMENTS

I want to first and foremost thank my supervisor, Alptekin Temizel, for his support and encouragement throughout my studies and for helping me conduct research independently. I am grateful to the members of my jury for their valuable recommendations and comments. I also want to thank my friends and colleagues in the METU Deep Learning and Computer Vision Lab group for their companionship. I want to express my sincere appreciation to my TÜBITAK 1001 team, which included Prof. Dr. Alptekin Temizel, Deniz Şen, Oguz Hanoğlu, and Ayberk Aydın. Collaborating with such a skilled team has been invaluable in helping me learn a lot in this domain, and contributed to the successful completion of my thesis work. Lastly, I am grateful to my mentor Ozan Kabak for the valuable guidance and support at work.

I would like to thank my family Berkan Bozkurt, Ayşe Yazgülü Güvercin and Ayşe Bozkurt who has been by my side throughout my whole life. I am extremely grateful to my parents Lütfiye Karlı and Mustafa Karlı who have raised me. Mom, you are the most magnificent mom I could have asked for, you have my deepest thanks. I also want to thank Mücteba for his friendship throughout this time. I want to give a special thanks to my girlfriend Selvi for supporting and encouraging me during this whole process whenever and wherever.

This work has been funded by The Scientific and Technological Research Council of Turkey, ARDEB 1001 Research Projects Programme project no: 120E093

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BIM	Basic Iterative Method
C&W	Carlini&Wagner
CAPTCHA	Completely Automated Public Turing test to tell Computers and Humans Apart
DNN	Deep Neural Networks
DISTS	Deep Image Structure and Texture Similarity
FGSM	Fast Gradient Sign Method
ILSVRC-2012	ImageNet Large Scale Visual Recognition Challenge 2012
JSMA	Jacobian-based Saliency Map Attack
L-BFGS	Limited Memory Broyden–Fletcher–Goldfarb–Shanno
LPIPS	Learned Perceptual Image Patch Similarity
PGD	Projected Gradient Descent
stAdv	Spatially Transformed Adversarial Examples

CHAPTER 1

INTRODUCTION

1.1 Problem Definition

In the last decade, there have been many improvements in Deep Neural Networks (DNNs) with great success. In 2012, a deep convolutional neural network was developed which was able to classify high-resolution images into 1000 different classes with a 15.3% test error rate and winning the ILSVRC-2012 competition as well as contributing to a series of breakthroughs in image classification [1]. Nowadays, these advances are being applied to real-life problems in various fields like speech processing [2], object detection [3], and gaming [4]. While DNNs are being used in a variety of domains, there are several studies that show vulnerabilities of DNNs against adversarial attacks. The initial study for generating adversarial examples was using the box-constrained L-BFGS method [5]. This study revealed that a DNN-based classifier could be deceived to misclassify an image by adding certain non-random perturbations.

As adversarial examples can fool the algorithms, they can be used for the purpose of distinguishing humans from algorithms. While humans could still perceive the content of these images, algorithms would be deceived by the adversarial input. Completely Automated Public Turing test to tell Computers and Humans Apart - CAPTCHA, is one of the most common applications where human users are distinguished from computer algorithms [6]. While humans could easily distinguish the input with or without perturbation, the adversarial perturbation fools the algorithms, so that human-accessible systems could be protected against unwanted digital access. CAPTCHA inputs could be divided into 4 main categories: text-based, image-based, audio-based, and video-based. State-of-the-art adversarial algorithms could be used to generate adversarial examples for text-based and image-based CAPTCHA [7] making them vulnerable to such attacks.

While DNNs are being used in a variety of domains, there are several studies that show their vulnerabilities. An initial study, L-BFGS method [5], revealed that neural networks are not robust to adversarial attacks specifically produced to fool the networks. After the discovery of adversarial attacks, several different methods have been proposed such as the Fast Gradient Sign Method (FGSM) [8], Projected Gradient Descent (PGD) [9], DeepFool [10], Jacobian Saliency Map Attack (JSMA) [11], Spatially Transformed Adversarial Examples (stAdv) [12] and Carlini&Wagner Attack [13].

Although adversarial examples could fool the networks, they may not preserve the perceptual quality. However, the perturbation that will be added to the image should be reasonably small so that human vision should not be distracted from the existence of perturbation, while networks are fooled by the adversary. Even though an adversarial attack would succeed in fooling the network, it is desired

that the attack does not reduce the perceptual quality of the image, i.e. fools the network with the smallest possible perturbation to the image. As one of the main motivations of this study is to find the least possible perceptual distortion, measurement of the perceptual distance is of critical importance. Despite the fact that there is no perceptual distance metric that exactly reflects human perception, there are several established perceptual metrics [14]. This study uses one traditional distance metric (i.e., L_2 norm) and one contemporary perceptual distance metric (i.e., LPIPS [14]). Consequently, this study proposes an ultimate integrated method composed of two separate methods to improve the perceptual quality of adversarial examples.

1.2 Contributions of the Study

The main motivation of this study is to improve the perceptual quality of successful adversarial attacks while reducing the perturbations that are distracting to humans. For this purpose, the integration of two separate methods is proposed to improve the perceptual quality while keeping the attacks successful. As these two methods are separate and applicable in different processing units, they could be integrated in a straightforward manner. While this ultimate method generates perceptually the best adversarial examples, primary and secondary methods could be applied separately to reduce the perceptual distance of any type of adversarial attack algorithm.

The primary method (i.e., Normalized Variance Weighting) is based on intensifying the perturbations, without any gradient calculations, in high-variance zones and suppressing in low-variance zones using the variance map of input images for any type of attack. In effect, disguising the adversarial noise in high-variance areas and limiting the high-frequency noise added to low-variance areas would be more distracting to the viewers.

The secondary method (i.e., Minimization) is based on the minimization of the perturbation until it reaches the decision boundary. While the primary method is applied during the attack, this method could be considered post-processing which is applied after the generation of an adversarial example with any type of adversarial attack. Localized and minimized perturbations improve the perceptual quality while keeping the fooling rate stable.

It has to be noted that this study does not propose new adversarial attack algorithms. Instead, this study proposes to perceptually improve any adversarial example that is generated by the existing adversarial attack algorithms while preserving the example as an adversary.

Preliminary findings of this work have been presented in the Adversarial Machine Learning and Beyond Workshop organized under the 36th AAAI Conference on Artificial Intelligence [15]:

Karli, B.T., Sen, D. and Temizel, A., 2021, November. Improving Perceptual Quality of Adversarial Images Using Perceptual Distance Minimization and Normalized Variance Weighting. In The AAAI-22 Workshop on Adversarial Machine Learning and Beyond.

1.3 Organization of the Thesis

- Chapter 2 presents the literature review with the existing methods for adversarial attack algorithms and perceptual metrics.

- Chapter 3 focuses on the methodology of this study, this chapter is divided into 3 subsections as follows: the Ultimate Method (integration of Normalized Variance Weighting and Minimization), the primary method (Normalized Variance Weighting Method), and the second method (Minimization Method).
- Chapter 4 shows the experiments with the applied datasets and attack types. This chapter presents and explains the results of the proposed primary and secondary methods besides the Ultimate Method.
- Chapter 5 discusses the proposed algorithms and the results.
- Chapter 6 concludes the study and mentions the possible future works.

CHAPTER 2

RELATED WORK

In this chapter, mainly the adversarial examples and perceptual metrics are examined. Adversarial examples are explained together with their application field, attack types (i.e., white-box, black-box), adversarial goals, attack types, and input requirements. After those properties of adversarial examples are presented, the most common adversarial attack algorithms in the literature are stated. Then, some recent perceptual metrics are mentioned besides the most common metrics.

2.1 Adversarial Examples

Adversarial examples could be divided into several subcategories in accordance with their application fields, attack types, and adversarial goals. Essentially, adversarial examples could be generated in various domains (e.g., Computer Vision, Natural Language Processing, Cyberspace Security or Attack in Physical World). All these different types of domains have subsections for the application field (e.g., Image Classification, Semantic Image Segmentation, and Object Detection for Computer Vision). Adversarial attacks could be divided into two subcategories according to their attack types: white-box attacks and black-box attacks. Adversarial goals, in a general view, could be divided into two main categories as targeted and non-targeted attacks. In a finer granularity, adversarial goals could be divided into 4 subcategories: Confidence Reduction, Misclassification, Targeted Misclassification, and Source/Target Misclassification [16]. Besides the adversarial output purposes (i.e., adversarial goals), adversarial attacks could also be categorized by input dependency. Adversarial algorithms could be divided into two subcategories regarding the input dependency of the models: Input Dependent Algorithms and Input Independent Algorithms. As mentioned above, there are various adversarial attack algorithms for different types of tasks, attack types, and adversarial goals.

2.1.1 Application Fields

There are several application fields for adversarial attacks. Some common application fields are mentioned below.

- **Computer Vision.** In this domain, adversarial examples could be applied to Image Classification [5] or Semantic Image Segmentation and Object Detection [17].
- **Natural Language Processing.** Text Classification [18] and Machine Translation [19] are some application fields in the this domain.

- **Cyberspace Security.** This subcategory has the following application fields: Cloud Service [20], Malware Detection [21] and Intrusion Detection [22].

- **Attacks in the Physical World.** There are several application areas of adversarial attacks that could be applicable in the physical world as follows: Spoofing Camera [23], Road Sign Recognition [24] or Face Recognition [25].

2.1.2 Attack Types

There are mainly two types of attack setups (i.e., White-box and Black-box).

- White-box attack setup assumes that the model, alongside all its parameters and data distributions are known. Obtaining the knowledge of the internal weights helps to produce confident adversarial examples [16].

- In the black-box attack setup, only the confidence levels or labels are known without any model information. This setup reflects more real-life scenarios regarding that accessing internal weight information is generally not possible in the practical examples [20].

2.1.3 Adversarial Goals

Adversarial goals could be divided into 4 as follows: [16]:

- Confidence reduction mainly aims to reduce the confidence score of the predicted output.

- Misclassification focuses on changing the prediction to any other class.

- Targeted Misclassification aims to change the output prediction to one specific target class for all inputs.

- Source/Target Misclassification aims to change the output prediction to a given list of specific target classes for each input.

2.1.4 Input dependency

- Input-dependent attacks focus on fooling each image separately. FGSM [8], C&W attack [13] or basic iterative method [26] are several examples of input-dependent adversarial attack algorithms.

- Input independent (i.e., universal) adversarial algorithms, introduced in [27], aim to generate one single perturbation (i.e., universal perturbation) that will fool the network when added to any image. There are several contemporary studies on universal adversarial perturbations (e.g., [28]).

2.1.5 Adversarial Attack Algorithms

There have been several adversarial attack algorithms developed since the discovery of the existence of adversarial examples in DNNs. L-BFGS is the initial method for generating adversarial examples using box-constrained optimization method [5]. However, this method is computationally very costly. FGSM [8] is an efficient gradient-based attack algorithm which computes the gradient only once, and adds perturbation in the gradient ascending direction of the loss function. Iterative Fast Gradient Sign Method (BIM) [26] extends FGSM by iteratively attacking with a small step size and calculating the gradient at each step. C&W Attack [13] minimizes L_2 norm with an improved optimization method. DeepFool [10] efficiently computes the smallest perturbations according to the closest decision boundary. Besides the input-dependent adversarial examples, Universal Adversarial Attacks produce input-independent universal adversarial examples [27].

2.1.5.1 L-BFGS

This study is the initial study that adversarial examples are introduced. L-BFGS is an optimization algorithm that is non-linear and gradient-based. The study describes the problem as an optimization problem and aims to show that adversarial examples could be obtained via optimization (i.e., with the L-BFGS optimization algorithm) [5]. Defining a classifier function $f : \mathbb{R}^m \rightarrow \{1...k\}$, which takes input image X and outputs the resulting class, $J : \mathbb{R}^m \times \{1...k\} \rightarrow \mathbb{R}^+$ denotes the loss function and l denotes target label $l \in \{1...k\}$. This attack intends to calculate r which minimizes the Equation 1 where $f(X + r) = l$ is satisfied.

$$Minimize\ c|r| + J(X + r, l)\ subject\ to\ X\ +\ r\ \in\ [0, 1]^m \tag{1}$$

2.1.5.2 FGSM

This study computes the loss of the gradient for X, calculates its negative of signs, and multiplies with the coefficient ϵ to produce X' where J is a cost function, X is the input, X' is the adversarial example and ϵ as the epsilon coefficient. As seen in Equation 2, an adversarial example is calculated in a single step without minimizing or iterating.

$$X' = X + \epsilon * sign(\nabla_x J(X, y_{true})) \tag{2}$$

While FGSM produces linear perturbations, this study shows that this approach also works on non-linear networks. Because of this linear nature, inexpensive, linear perturbations of a linear model are simply enough to harm the networks [8].

2.1.5.3 BIM

This study which is called as basic iterative method proposes an iterative version of FGSM with smaller steps, α. In every iteration (where N is the iteration count), the algorithm applies the FGSM algorithm. As seen in the Equation 3, this method clips the perturbation to (ϵ, -ϵ) intervals after each iteration [26].

$$X_0^{adv} = X, X_{N+1}^{adv} = Clip_x, \epsilon(\alpha * sign(\nabla_x J(X_N^{adv}, y_{true})) \tag{3}$$

2.1.5.4 DeepFool

Deepfool assumes that there is a hyperplane for each class. The algorithm aims to find the closest hyperplane and projects input onto the plane so that the adversary could cross the decision boundary. This iterative process focuses on finding the minimum perturbation that is required with respect to the decision boundaries [10].

2.1.5.5 Carlini&Wagner Attack

Carlini&Wagner Attack reformulates the optimization problem that was introduced in L-BFGS adversarial attack. In Equation 4, c is a chosen constant calculated with binary search and δ denotes the perturbation. The purpose of the formulation of δ is to smooth over the clipped gradient descent. This study proposes 3 different methods based on 3 different distance norms (i.e., L_2, L_0, and L_∞ norms).

$$\begin{aligned} minimize\ ||\delta||_p + c * f(x+\delta) \\ such\ that\ x+\delta \in [0,1]^n \\ where\ \delta_i = 1/2 * (\tanh(w_i) + 1) - x_i \end{aligned} \tag{4}$$

2.1.5.6 Universal Adversarial Perturbations

This study shows that input-independent universal perturbations could be effective across neural networks. Additionally, universal adversarial perturbations reveal geometric relations about the networks' decision boundaries [27].

2.1.6 Adversarial Attacks Using Maps

Some adversarial attack algorithms apply different types of mappings for different purposes while generating adversarial examples (e.g., Saliency maps and Variance maps). Jacobian-based Saliency Map Attack (JSMA) [11] generates sparse perturbations via generating a saliency map. Besides, variance map-based studies [29, 30] have been studied to improve perceptual quality in previous works.

2.1.6.1 Saliency Maps on Adversarial Examples

These types of attacks mostly focus on minimizing the L_0 norm. L_0 norm-based adversarial attacks could fool the networks by only perturbing a small portion of the input variables. JSMA computes the gradient for each input variable to produce a saliency map. After generating the saliency map, the algorithm ranks the contribution of each input variable to the adversarial objective. A perturbation is then selected from the saliency map at each iteration [11].

2.1.6.2 Variance Maps on Adversarial Examples

Human perception is affected more by perturbations in the low variance areas compared to high variance areas and this information is exploited in various image processing applications [31, 32, 33]. Regarding this fact, a variance map has been used in previous studies [29, 30] to generate adversarial examples. In one study, a variance map is used to produce variance-based component-wise box constraints to generate sparse adversarial examples [30]. In another study, a variance map is applied for the selection of high variance pixels [29]. Using only L_p norms for these variance-based sparse attacks does not accurately reflect the perceptual quality [29], thus variance based sparse adversarial examples either use mean and median values of pixels or introduce a new distance metric that is more suited for the evaluation of their proposed attacks.

2.2 Perceptual Metrics

All adversarial attack methods essentially aim to fool the network while minimizing the dissimilarity between benign and adversarial examples (i.e., minimizing the added perturbation). While the similarity metrics vary according to attack type, the most widely used distance metrics are L_p norms.

However, there are some attack types such as [34, 35, 36, 37] for which L_p norms are not fully suitable to evaluate the attack success. Thus these studies employ different and more recent perceptual metrics such as Learned Perceptual Image Patch Similarity (LPIPS) metric [14] or Deep Image Structure and Texture Similarity (DISTS) index [38]. Both of these methods use an additional neural network to measure the distance.

2.2.1 L_p Norms

L_p norms are the most common distance metrics and generally, 4 basic L_p norms are applied (i.e., p = 0, 1, 2 and ∞). In particular, FGSM is an L_∞, JSMA is an L_0, and C&W is an L_2 norm-based attack. Even though L_p norms are very convenient and commonly used, several studies state that L_p norms do not reflect human perception accurately [39, 34].

2.2.2 LPIPS

LPIPS calculates the similarity between given two images or batches by measuring the Euclidean distance of deep representations. This study shows that DNNs could be used as a perceptual distance

metric independent of network type. At the final stage, LPIPS is calibrated with human perception by integrating an additional linear layer to its core network [14].

2.2.3 DISTS

DISTS computes the similarity between the main image and the degraded version of the image. DISTS contains a convolutional neural network that is optimized over human perception while using the combination of deep image structure and texture similarity [38].

CHAPTER 3

METHODOLOGY

In this section, we first describe the proposed Normalized Variance Weighting, which is the primary method. Then, we describe the Minimization Method which is the secondary. Finally, we introduce an ultimate method which is the combination of Normalized Variance Weighting and Minimization methods. Note that all proposed methods are attack agnostic methods aiming to increase the perceptual similarity between the input and the adversarial example.

3.1 Normalized Variance Weighting

In this study, a variance map is used as a guide to intensify the perturbations in the high-variance zones. For this purpose, the variance map method in [30] is used to produce a variance map of input images.

In this method, the standard deviation of both axes with 2 neighbor pixels and the main pixel for each color channel are calculated ($\sigma_{ij}^{(x)}$ and $\sigma_{ij}^{(y)}$ for x and y axis respectively) and the square root of the minimum of the standard deviation of axes is taken to obtain variance map σ_{ij} (Equation 5). The variance map is then normalized to obtain normalized variance map $V_{i,j}$ (Equation 6).

$$\sigma_{ij} = \sqrt{\min\left\{\sigma_{ij}^{(x)}, \sigma_{ij}^{(y)}\right\}} \tag{5}$$

$$V_{i,j} = \frac{\sigma_{i,j}}{\sqrt{\sum_{h}^{H}\sum_{w}^{W}\sigma_{h,w}^{2}}} \tag{6}$$

Since the proposed method of this study does not involve selecting pixels or generating variance box constraints, it does not require any additional threshold or coefficient variable. Normalizing and weighting procedures remove the need for an additional variable. As seen in Algorithm 1, the proposed method initially generates the variance map of input images only once (Equation 5), then normalizes the variance map using L_2-norm (Equation 6). Finally, the algorithm applies the variance map by weighting the perturbation (i.e., element-wise multiplication of normalized variance map and the perturbation) with normalized variance map at each iteration (if an adversarial attack is iterative) as seen in the Equation 7 where x is the benign sample, y is the adversarial sample, δ is the initial perturbation and δ_{var} is the resulting perturbation after the Normalize Variance Weighting Method. This method could be adapted for both white-box and black-box settings and does not require any optimization, or

Algorithm 1 Normalized Variance Weighting

Input: x: original image, Adv: one iteration of adversarial attack
Parameter: i_{max}: maximum iteration of adversarial attack
Output: y: adversarial example

Let $i = 0$
$v = VarianceMap(x)$
$v = L_2Normalize(v)$
$y = x$
while $i < i_{max}$ **do**
  $y = Adv(y)$
  $p = (y - x) \times v$
  $y = x + p$
  $i = i + 1$
end while
return y

additional gradient-based steps. Therefore it does not bring any additional computational cost except the calculation of the variance map for once and one matrix multiplication (i.e., multiplication of perturbation with the Normalized Variance Map) at each iteration. Since variance-weighted perturbations are applied with normalization, this method is not expected to create any significant changes in L_2 distance. However, the LPIPS distance [14] metric, which is calibrated by human perception, would be a better metric to measure the perceptual enhancements for the Normalized Variance Weighting Method.

$$\begin{aligned} y &= Adv(x) \\ \delta &= y - x \\ \delta_{var} &= \delta \odot V \end{aligned} \tag{7}$$

3.2 Minimization Method

The proposed minimization method is applied after generating the initial successful adversarial sample and aims to reduce the distance between benign and adversarial examples using an optimizer. Optimizer minimizes the distance until adversarial examples reach the decision boundaries of true classes or maximum iteration number (Algorithm 2).

$$\begin{aligned} y &= Adv(x) \\ minimize\ &||y - x||_n \\ such\ that\ &x_{label} \neq y_{label} \\ where\ &n \in [LPIPS, L_2] \end{aligned} \tag{8}$$

It is very crucial to initially produce a successful adversarial example for this method considering this algorithm only promises to keep the adversary, and does not guarantee to produce one. After acquiring

a successful adversary, the algorithm focuses on minimizing the dissimilarity of the benign and the adversarial sample (Equation 8).

There could be several metrics used to measure and minimize the dissimilarity between benign and adversarial examples. The proposed minimization technique is applied to minimize with regards to two different metrics: L_2-norm and LPIPS (It has to be noted that some attacks are not fully suitable for L_2 distance metric [37]). However, unlike conventional norms, LPIPS has a higher computational cost compared to L_2-norm minimization since LPIPS measures the perceptual distance using an additional neural network (i.e., VGG16 [40]). While LPIPS internally contains an additional neural network we were able to minimize the LPIPS distance since the implementation of the LPIPS distance metric is compatible to backpropagate through its loss. In such a manner, it is shown that contemporary and more complex distance metrics could also be used to form the adversary.

In [6], the proposed algorithm in this work also minimizes the perturbation independent of the adversarial attack. However, in this method the attack strength is iteratively adjusted to obtain the minimal perturbation needed in an attack-agnostic manner after the generation of an adversarial example, our proposed method improves this by directly optimizing the minimization of perturbation.

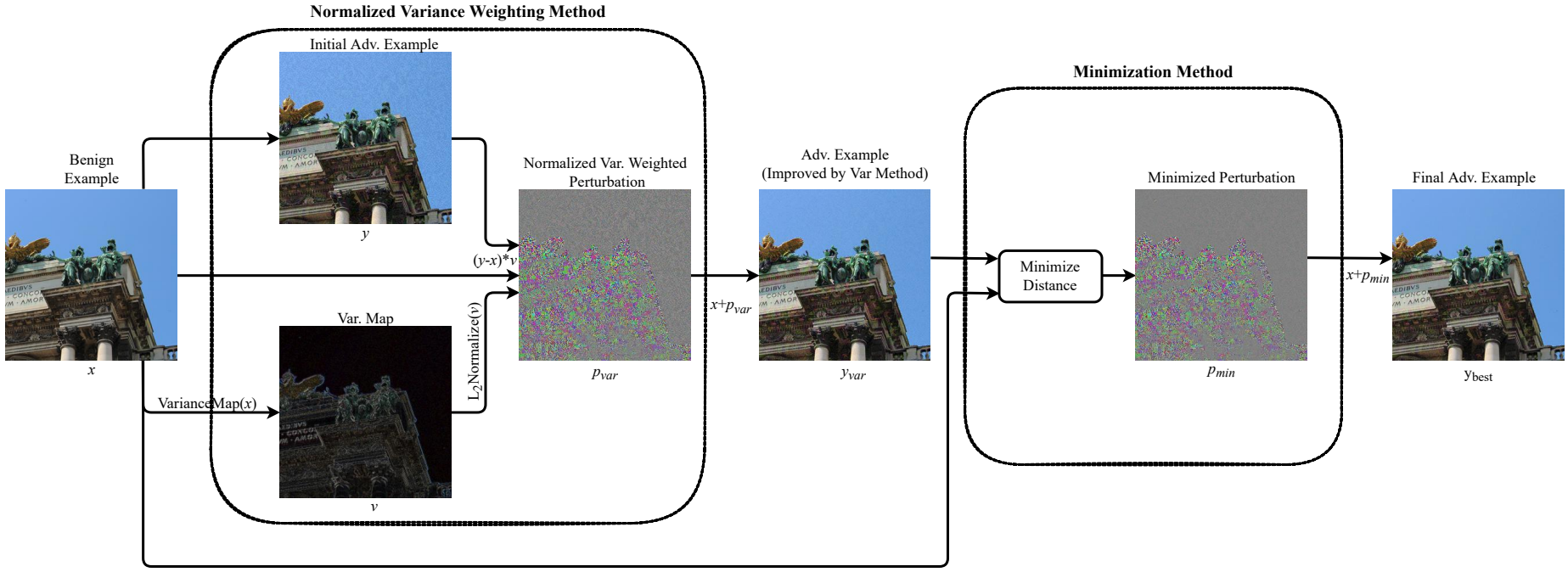


Figure 1: Visual representation of the integration of proposed methods.

Algorithm 2 Minimization Method

Input: x: original image, y: adversarial example
Parameter: lr: learning rate, i_{max}: maximum iteration
Output: y_{best}: improved adversarial example

Let $i = 0$
$y_{best} = y$
$y_{opt} = y$
while $i < i_{max}$ **do**
  if $class_{y_{opt}} == class_x$ **then**
    return y_{best}
  else
    $y_{best} = y_{opt}$
  end if
  $y_{opt} = MinimizeDIST(y_{opt}, x, lr)$
end while
return y_{best}

3.3 Ultimate Method

The ultimate Method is composed of the Normalized Variance Weighting and Minimization Method. Normalized Variance Weighting is applied during the adversarial attack while Minimization Method is applied after the generation of an adversarial example. Therefore, both methods could be integrated and used in association with the generation of adversarial examples (Algorithm 3).

The complete pipeline integrating both methods is illustrated in Figure 1. Firstly the variance map of the image(s) will be generated, then the variance weighting method will be applied at each iteration of the regular adversarial attack algorithm for iterative attacks. After the generation of variance-weighted adversarial examples, the Minimization Method will be applied as post-processing to obtain perceptually more improved adversarial examples.

(a) Benign Example (b) Vanilla Attack

(c) Var. Map (d) Var. Method

(e) L_2-Minimization (f) LPIPS-Minimization

(g) Var. + L_2-Minimization (h) Var. + LPIPS-Minimization

Figure 2: FGSM ($\epsilon = 12/255$) results on NIPS2017 against ResNet50. a) Benign example. b) FGSM result. c) Variance map. d) Adv. example improved by variance method. e) Adv. example improved by L_2-Minimization. f) Adv. example improved by LPIPS-Minimization. g) Adv. example improved by Var. + L_2-Minimization. h) Adv. example improved by Var. + LPIPS-Minimization.

(a) Benign Example

(b) Vanilla Attack

(c) Var. Map

(d) Var. Method

(e) L_2-Minimization

(f) LPIPS-Minimization

(g) Var. + L_2-Minimization

(h) Var. + LPIPS-Minimization

Figure 3: BIM ($\epsilon = 8/255$, $\alpha = 0.8/255$, it = 10) results on NIPS2017 against ResNet50. a) Benign example. b) BIM result. c) Variance map. d) Adv. example improved by Var. method. e) Adv. example improved by L_2-Minimization. f) Adv. example improved by LPIPS-Minimization. g) Adv. example improved by Var. + L_2-Minimization. h) Adv. example improved by Var. + LPIPS-Minimization.

Algorithm 3 Normalized Variance Weighting + Minimization

Input: x: original image, Adv: one iteration of adversarial attack

Parameter: lr: learning rate, i_{max}: maximum iteration

Output: y_{best}: improved adversarial example

Let $i = 0$
$v = VarianceMap(x)$
$v = L_2Normalize(v)$
$y = x$
while $i < i_{max}$ **do**
    $y = Adv(y)$
    $p = (y - x) \times v$
    $y = x + p$
    $i = i + 1$
end while
$y_{best} = y$
$y_{opt} = y$
while $i < i_{max}$ **do**
    if $class_{y_{opt}} == class_x$ **then**
        return y_{best}
    else
        $y_{best} = y_{opt}$
    end if
    $y_{opt} = MinimizeDIST(y_{opt}, x, lr)$
end while
return y_{best}

CHAPTER 4

EXPERIMENTAL EVALUATION

This chapter provides details of the experimental settings and results. Under the Experimental Settings section, applied datasets and attack types are explained. Following this section, experimental settings for the Normalized Variance Weighting and Minimization Method are given. Finally, experimental results are examined for Normalized Variance Weighting, the Minimization Method, and the Ultimate Method respectively.

4.1 Experimental Settings

4.1.1 Datasets

CIFAR-10 and NIPS2017 Adversarial Learning Challenge datasets are used in the experiments. CIFAR-10 test set contains 10000 images with 32×32 resolution. Experiments are conducted on a subset of CIFAR-10 test set with 1000 images (100 random images from each category). NIPS2017 dataset is a subset of Imagenet dataset and contains 1000 images (one image from each category) with 299×299 resolution.

4.1.2 Attack Types

Proposed methods have been tested using 3 different non-targeted attack types: a single step gradient based attack (FGSM) [8], an iterative gradient-based attack (BIM) [26] and an optimization-based attack (C&W) [13] on CIFAR10 and NIPS2017 datasets. ResNet50 [41] and Inception-V3 [42] have been used for NIPS2017 dataset; only ResNet50 [41] for CIFAR10 dataset. CleverHans [43] implementation for default attacks is applied and the proposed methods are integrated into these attacks.

4.1.3 Experimental Settings for Normalized Variance Weighting

For the variance map, 2 neighbor pixels and the main pixel are used for every color channel in the generation of variance map similar to the study Sparse and Imperceivable Adversarial Attacks [30]. It has been observed that using the variance weighting method considerably decreases the fooling rate when the attack strength is fixed. Thus, to compare on the fairground, the fooling rates are fixed at given percentages and ϵ (for FGSM and BIM attacks) or initial cost (for C&W attack) is allowed to

vary. This makes it possible to reach the target fooling rate within a $\pm 0.5\%$ error tolerance. 4 different fooling rates are targeted for FGSM: 30%, 40%, 50%, 60%, and BIM: 60%, 70%, 80%, 90% on both datasets. A single fooling rate has been used for C&W attack on each dataset: 95% on CIFAR10 and 100% on NIPS2017 (for both ResNet50 and Inception-V3), since there is L_2-normalization after producing a variance map, measuring L_p norms would be misleading for variance weighting method. Therefore, mainly LPIPS perceptual distance metric is considered, which is calibrated with human vision, for its evaluation. Those experiments have been conducted in a workstation with two NVIDIA RTX 3080 GPUs.

4.1.4 Experimental Settings for Minimization Method

For the experimental settings of the proposed minimization method, Adam [44] is used as the optimizer and the maximum iteration number is set to 10. The learning rate is set as 0.0001 for the CIFAR10 dataset for both minimization methods. The learning rate is set as 0.0001 for L_2-Minimization and 0.00001 for LPIPS-minimization on NIPS2017 (for both ResNet50 and Inception-V3) since they were not converging with the same learning rate. Also, those experiments have been conducted in a workstation with two NVIDIA RTX 3080 GPUs.

4.2 Results

4.2.1 Normalized Variance Weighting

The effect of the variance weighting method on FGSM attack can be observed in Table 1 and Table 5 for CIFAR10 (using ResNet50) and NIPS2017 (using ResNet50 and Inception-V3) datasets respectively. The method reduces the LPIPS distances considerably in all cases, i.e. without minimization and when used together with minimization with respect to both L_2 and LPIPS. The corresponding results in Table 2 and Table 6 for BIM and Table 3 and Table 4 for C&W attack confirms that these findings are pertinent to these attacks as well and variance weighting is effective in reducing the LPIPS distance for all attack types in question.

More specifically Variance Weighting Method reduces the LPIPS distances by around 40% (from 0.0093 to 0.0054 for fooling rate 40%) on the CIFAR10 dataset (Table 1) and 43% (from 0.0014 to 0.0008 for fooling rate 40%) on NIPS2017 dataset (Table 5) for FGSM attack and around 10% (from 0.0097 to 0.0087 for fooling rate 70%) on CIFAR10 dataset (Table 6) and 29% (from 0.0017 to 0.0013 for fooling rate 70%) on NIPS2017 dataset (Table 2) for the BIM attack with little computational overhead. The method improves LPIPS perceptual distance of C&W attack by around 4% (from 0.0054 to 0.0052) on CIFAR10 (Table 3) and 24% (from 0.0025 to 0.0019) on NIPS2017 (Table 4). On the other hand, as expected, there is no improvement in L_2 distances.

4.2.2 Minimization Methods

The LPIPS-Minimization method applied on FGSM attack decreases the LPIPS distances considerably when used individually as well as when it is combined together with variance weighting for both CIFAR10 (using ResNet50) and NIPS2017 (using ResNet50 and Inception-V3) datasets (Table 1 and

Table 1: FGSM results on CIFAR10 dataset against ResNet50 with and without variance weighting (shown as Var.) and minimization method (shown as Minim.) using LPIPS and L_2. Results are reported in both LPIPS ($\times 10^2$) and L_2 metrics.

Fixed Fooling Rates:		30%		40%		50%		60%	
Var.	Minim.	LPIPS	L_2	LPIPS	L_2	LPIPS	L_2	LPIPS	L_2
-	-	0.19	0.13	0.93	0.28	3.19	0.58	7.42	1.14
-	LPIPS	0.07	0.13	0.59	0.27	2.58	0.57	6.46	1.13
-	L_2	0.10	**0.09**	0.75	**0.22**	2.93	0.52	7.06	**1.09**
+	-	0.16	0.14	0.54	0.26	2.36	0.54	6.85	1.18
+	LPIPS	**0.06**	0.13	**0.29**	0.25	**1.82**	0.54	**5.96**	1.17
+	L_2	0.09	0.10	0.41	**0.22**	2.14	**0.50**	6.54	1.13

Table 2: BIM results on CIFAR10 dataset against ResNet50 with and without variance weighting (shown as Var.) and minimization method (shown as Minim.) using LPIPS and L_2. Results are reported in both LPIPS ($\times 10^2$) and L_2 metrics.

Fixed Fooling Rates:		60%		70%		80%		90%	
Var.	Minim.	LPIPS	L_2	LPIPS	L_2	LPIPS	L_2	LPIPS	L_2
-	-	0.62	0.26	0.97	0.33	1.52	0.41	2.71	0.57
-	LPIPS	0.28	0.25	0.50	0.32	0.86	0.40	1.74	0.56
-	L_2	0.43	**0.21**	0.73	**0.28**	1.22	**0.37**	2.32	**0.52**
+	-	0.57	0.27	0.87	0.34	1.37	0.43	2.51	0.60
+	LPIPS	**0.26**	0.26	**0.43**	0.33	**0.76**	0.42	**1.58**	0.59
+	L_2	0.41	0.23	0.66	0.30	1.11	0.39	2.16	0.56

Table 5). The corresponding results in Table 2 and Table 6 for BIM and Table 3 and Table 4 for C&W attack confirm that these findings are pertinent to these attacks as well and LPIPS is effective in reducing the LPIPS distance for all attack types in question.

Separately LPIPS-Minimization method decreases the LPIPS distances around 37% (from 0.0093 to 0.0059 for fooling rate 40%) on the CIFAR10 dataset (Table 1) and 79% (from 0.0014 to 0.0003 for fooling rate 40%) on NIPS2017 dataset (Table 5) for FGSM attack. Likewise, the LPIPS-Minimization method reduces the LPIPS distances around 49% (from 0.0097 to 0.0050 for fooling rate 70%) on the CIFAR10 dataset (Table 2) and 82% (from 0.0017 to 0.003 for fooling rate 70%) on NIPS2017 dataset (Table 6) for BIM attack. Additionally, the LPIPS-Minimization method decreases LPIPS perceptual distance around 39% (from 0.0054 to 0.0033) on the CIFAR10 dataset (Table 3) and 40% (from 0.0025 to 0.0015) on NIPS2017 dataset (Table 4) for C&W attack.

In addition to these results, the LPIPS-Minimization method also improves L_2 distance considerably. Though, as expected, the L_2-Minimization method results in the best L_2 distance improvements for FGSM and BIM on both CIFAR10 (using ResNet50) and NIPS2017 (using ResNet50 and Inception-V3) datasets. Considering C&W is already optimizing L2 distance, the improvement is relatively limited for C&W attack on both datasets.

Table 3: CW2 Results on CIFAR10 dataset against ResNet50 with and without variance weighting (shown as Var.) and minimization method (shown as Minim.) using LPIPS and L_2. Results are reported in both LPIPS ($\times 10^2$) and L_2 metrics.

Var.	Minim.	LPIPS	L_2
-	-	0.25	0.27
-	LPIPS	0.15	0.27
-	L_2	0.25	0.28
+	-	0.19	0.27
+	LPIPS	**0.12**	**0.26**
+	L_2	0.19	0.28

Table 4: CW2 Results on NIPS2017 dataset against ResNet50 and Inception-V3 with and without variance weighting (shown as Var.) and minimization method (shown as Minim.) using LPIPS and L_2. Results are reported in both LPIPS ($\times 10^2$) and L_2 metrics.

Models:		ResNet50		Inc-V3	
Var.	Minim.	LPIPS	L_2	LPIPS	L_2
-	-	0.25	0.27	0.33	0.38
-	LPIPS	0.15	0.27	**0.17**	**0.37**
-	L_2	0.25	0.28	0.32	0.38
+	-	0.19	0.27	0.33	0.45
+	LPIPS	**0.12**	**0.26**	0.19	0.45
+	L_2	0.19	0.28	0.32	0.46

L_2-Minimization method decreases the L_2 distance by about 21% (from 0.28 to 0.22 for fooling rate 40%) on CIFAR10 dataset Table 1 and 34% (from 0.32 to 0.21 for fooling rate 40%) on NIPS2017 Table 5 for FGSM attack. L_2 distances are reduced by 15% (from 0.33 to 0.28 for fooling rate 70%) on CIFAR10 (Table 2) and 25% (from 0.40 to 0.30 for fooling rate 70%) on NIPS2017 Table 6 for BIM attack. Table 3 shows that L_2 distance is decreased by 6% (from 0.18 to 0.17) on CIFAR10 and while there is no improvement on NIPS2017 (Table 4).

4.2.3 Ultimate Method

Integration of the Normalized Variance Weighting Method and the LPIPS-Minimization Method reduces the LPIPS distance by about 69% (from 0.0093 to 0.0029 for fooling rate 40%) on the CIFAR10 dataset (Table 1) and 86% (from 0.0014 to 0.0002 for fooling rate 40%) on NIPS2017 dataset (Table 5) for FGSM attack. This integrated method decreases LPIPS distance around 56% (from 0.0097 to 0.0043 for fooling rate 70%) on the CIFAR10 dataset (Table 2) and 82% (from 0.0017 to 0.0003 for fooling rate 70%) on NIPS2017 dataset (Table 6) for the BIM attack. LPIPS distance is reduced by 41% (from 0.0054 to 0.0032) on CIFAR10 (Table 3) and 52% (from 0.0025 to 0.0012) on NIPS2017 (Table 4) for C&W attack. According to these results, the minimum LPIPS perceptual distances are obtained by this method and minimum L_2 distances are obtained by the L_2-Minimization method.

The best results (i.e. lowest LPIPS distances) are obtained when Variance Weighting Method with LPIPS-Minimization is combined. As could be seen from experimental results, quantitatively there is a considerable improvement for FGSM, BIM, and C&W attack types on both CIFAR10 (using ResNet50) and NIPS2017 (using both ResNet50 and Inception-V3). The empirical visual improvements of FGSM and BIM attack algorithms could also be seen in Figure 2 and 3.

Table 5: FGSM results on NIPS2017 dataset against ResNet50 and Inception-V3 with and without variance weighting (shown as Var.) and minimization method (shown as Minim.) using LPIPS and L_2. Results are reported in both LPIPS ($\times 10^2$) and L_2 metrics.

Fixed Fooling Rates:		30%				40%				50%				60%			
Models:		ResNet50		Inc-V3		ResNet50		Inc-V3		ResNet50		Inc-V3		ResNet50		Inc-V3	
Var.	Minim.	LPIPS	L_2	LPIPS	L_2	LPIPS	L_2	LPIPS	L_2	LPIPS	L_2	LPIPS	L_2	LPIPS	L_2	LPIPS	L_2
-	-	0.07	0.22	0.12	0.30	0.14	0.32	0.26	0.43	0.26	0.42	0.54	0.63	0.44	0.55	1.10	0.94
-	LPIPS	0.02	0.20	0.02	0.27	0.03	0.28	0.04	0.39	0.05	0.38	0.10	0.57	0.08	0.50	0.26	0.87
-	L_2	0.05	0.16	0,07	0,20	0.10	0.21	0,11	**0,24**	0.13	**0.25**	0,22	**0,37**	0.23	**0.35**	0,30	**0,39**
+	-	0.04	0.26	0,06	0,35	0.08	0.37	0,13	0,49	0.14	0.49	0,25	0,71	0.22	0.62	0,52	1,04
+	LPIPS	**0.01**	0.23	**0,01**	0,32	**0.02**	0.34	**0,02**	0,45	**0.03**	0.46	**0,05**	0,66	**0.05**	0.58	**0,13**	0,99
+	L_2	0.02	**0.15**	0,03	**0,17**	0.04	**0.20**	0,05	0,26	0.08	0.29	0,09	0,38	0.12	0.39	0,15	0,56

Table 6: BIM Results on NIPS2017 dataset against ResNet50 and Inception-V3 with and without variance weighting (shown as Var.) and minimization method (shown as Minim.) using LPIPS and L_2. Results are reported in both LPIPS ($\times 10^2$) and L_2 metrics.

Fixed Fooling Rates:		60%				70%				80%				90%			
Models:		ResNet50		Inc-V3		ResNet50		Inc-V3		ResNet50		Inc-V3		ResNet50		Inc-V3	
Var.	Minim.	LPIPS	L_2	LPIPS	L_2	LPIPS	L_2	LPIPS	L_2	LPIPS	L_2	LPIPS	L_2	LPIPS	L_2	LPIPS	L_2
-	-	0.11	0.32	0.23	0.43	0.16	0.38	0.37	0.56	0.23	0.46	0.63	0.75	0.40	0.63	1.19	1.08
-	LPIPS	0.02	0.28	0.04	0.39	0.03	0.34	0.07	0.51	0.04	0.42	0.12	0.69	0.07	0.58	0.28	1.01
-	L_2	0.07	**0.22**	0.14	**0.29**	0.12	**0.28**	0.21	**0.36**	0.18	**0.36**	0.25	**0.44**	0.25	**0.45**	0.32	**0.55**
+	-	0.09	0.41	0.13	0.52	0.12	0.49	0.21	0.67	0.18	0.61	0.37	0.90	0.31	0.81	0.71	1.29
+	LPIPS	**0.02**	0.38	**0.03**	0.47	**0.03**	0.45	**0.04**	0.62	**0.04**	0.56	**0.08**	0.84	**0.06**	0.76	**0.17**	1.23
+	L_2	0.06	0.28	0.08	0.35	0.09	0.36	0.13	0.45	0.13	0.47	0.15	0.59	0.18	0.61	0.20	0.78

CHAPTER 5

DISCUSSION

This chapter initially discusses the experimental and empirical observations of the proposed methods. Then alternative methods for Normalized Variance Weighting are discussed together with their advantages and limitations.

5.1 Experimental and Empirical Observations

Experimental and empirical observations are discussed in the following section. Firstly, limitations and edge cases are mentioned, then observations on applied metrics are stated.

5.1.1 Limitations

From the experiments, it is seen that both variance and minimization methods individually improve the perceptual quality and the best LPIPS results are obtained when the Ultimate Method is applied. However, this improvement is relatively limited for attacks that are inherently able to produce adversarial examples with quantitatively lower perceptual distances, such as C&W. Results in Table 1 to 4 show that NIPS2017 dataset results have smaller perceptual distances yet improvement percentages are higher compared to CIFAR10 dataset (i.e., LPIPS distance is reduced by 10% and 25% for CIFAR10 (40% fooling rate) and NIPS2017 (70% fooling rate) datasets respectively for BIM against ResNet50).

Our empirical observations show that the variance weighting method significantly improves the perceptual quality of images that have partially low variance backgrounds (e.g., sky, wall, or sea) as could be seen in Figure 2, 3. However it is less effective for images with dominantly high variance zones (e.g., umbrella image in Figure 4(b)). If an image has uniformly high variance or low variance pixels, then the variance map will become a matrix of ones that will have no effect on adversarial examples. However, the perceptual quality may even worsen in images containing dominantly low variance areas with only a small portion of high variance areas (e.g., flag image in Figure 4(b)). The flag image is almost completely composed of low variance zones considering the background while the high variance region is highly narrow. Hence variance weighting method could not improve the flag image adequately and it even worsens the perceptual quality of the adversarial example.

(a) Umbrella Vanilla Adv.

(b) Umbrella Variance Adv.

(c) Flag Vanilla Adv.

(d) Flag Variance Adv.

Figure 4: Unimproved NIPS2017 samples against Normalized Variance Weighting. a) FGSM Adv. example with dominantly high variance pixels. b) Adv. example after the variance method is applied. c) FGSM Adv. example with dominantly low variance pixels. d) Adv. example after the variance method is applied.

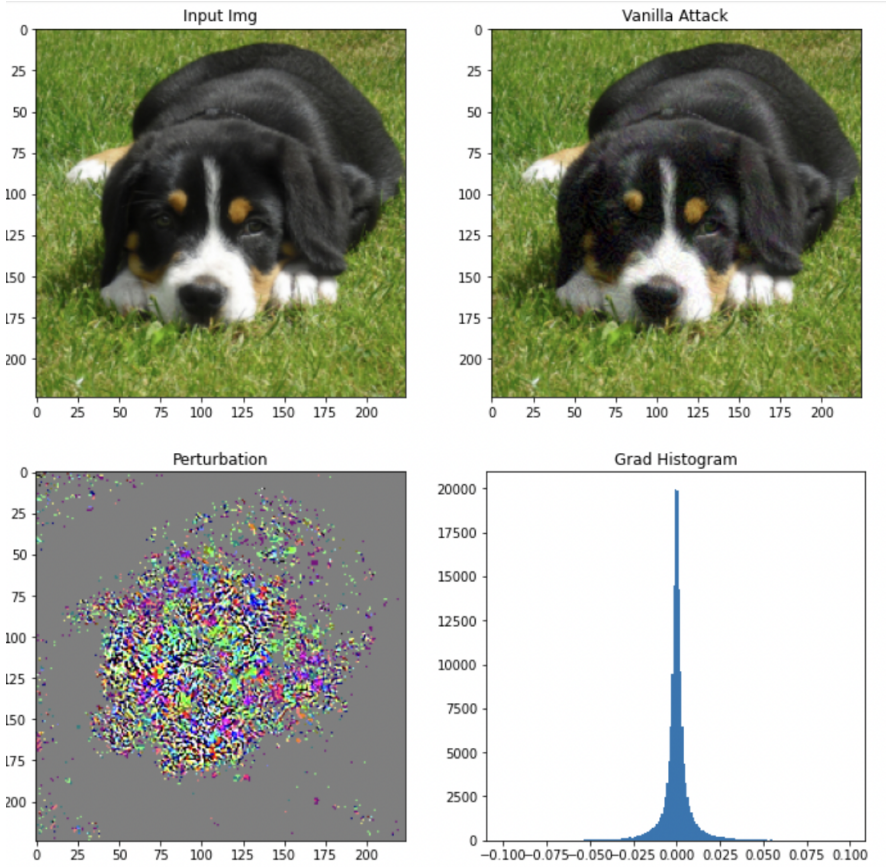


Figure 5: Quantile Gradient Norm Masking Method.

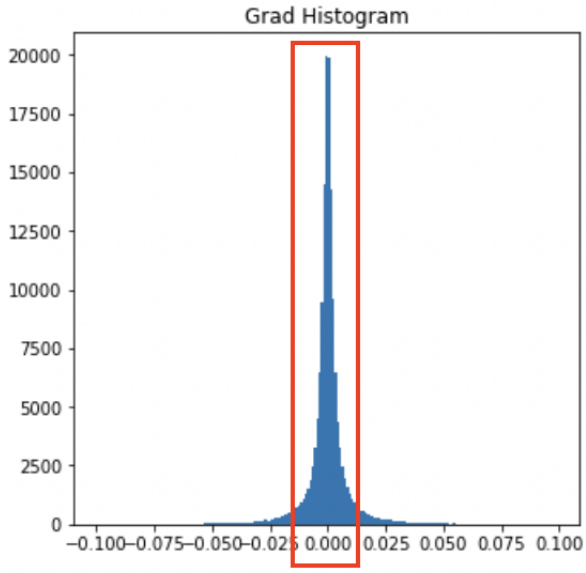
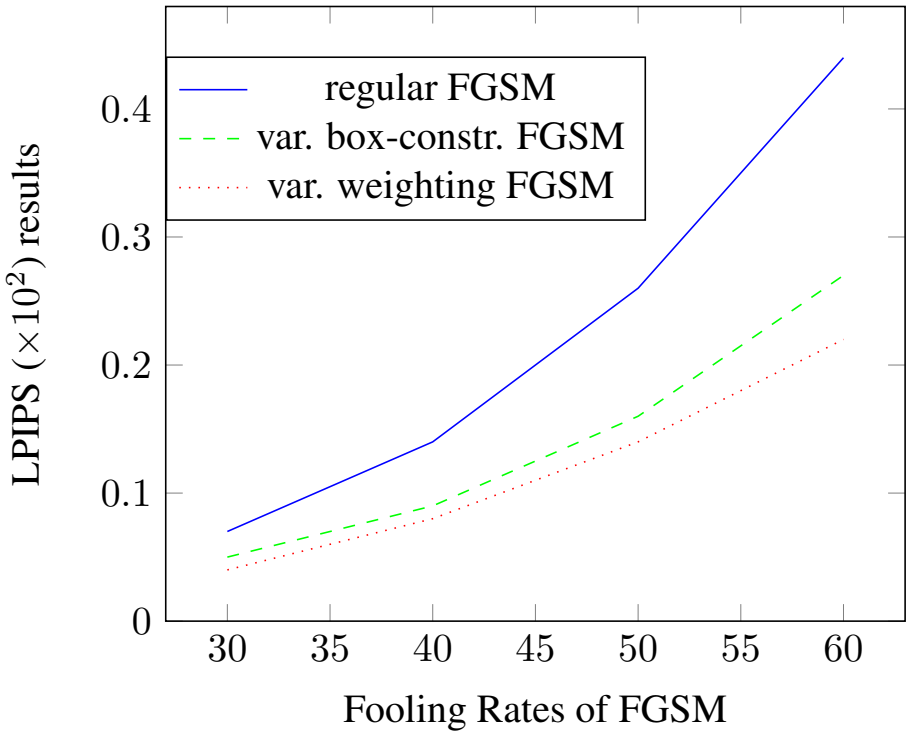
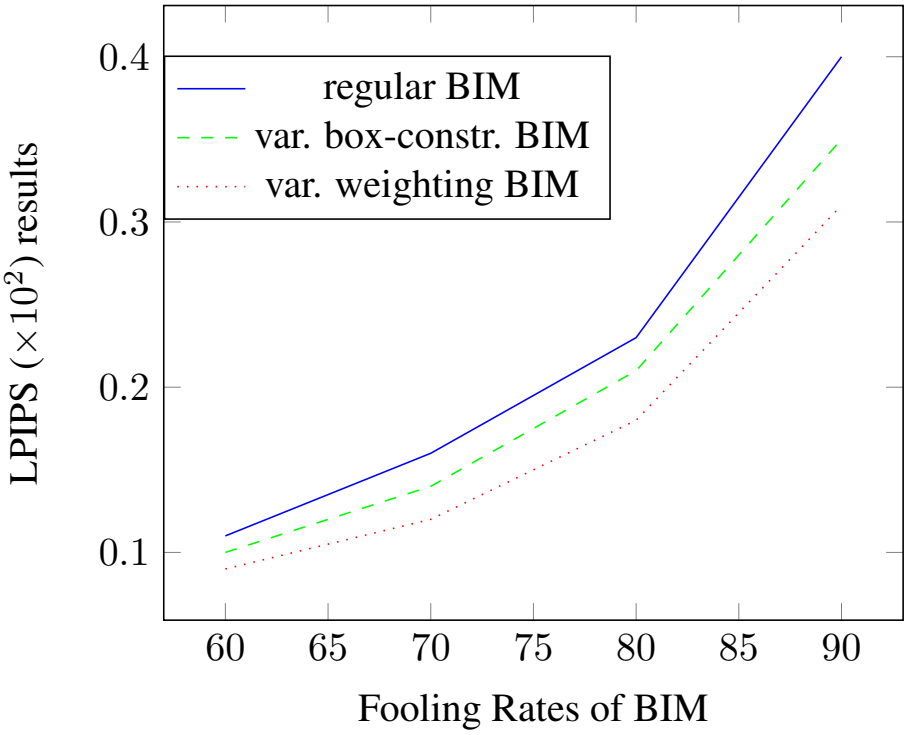


Figure 6: Histogram of the Gradient Norm with for the selected quantile q=0.7.



(a) Comparison of improvement methods on FGSM



(b) Comparison of improvement methods on BIM

Figure 7: LPIPS results on NIPS2017 dataset with regular method, Normalized Variance Weighting Method, and variance-based box-constrained method using ResNet50. a) Results of FGSM attack. b) Results of BIM attack.

Table 7: FGSM attack results against ResNet50 with Quantile Gradient Norm Masking Method (Epsilon ϵ = 0.005, Quantile q = 0.7)

Grad. Mask	Minimization	L_0	L_1	L_2	L_∞	Fooling Rate
-	-	3072.00	15.36	0.28	0.005	0.35
-	+	3072.00	12.85	0.23	0.005	0.35
+	-	**2150.00**	10.75	0.23	0.005	0.35
+	+	2804.28	**9.45**	**0.19**	0.005	0.35

Table 8: FGSM attack results against ResNet50 with Quantile Gradient Norm Masking Method (Epsilon ϵ = 0.05, Quantile q = 0.7)

Grad Mask	Minimization	L_0	L_1	L_2	L_∞	Fooling Rate
-	-	3072.00	153.60	2.77	0.05	0.65
-	+	3072.00	151.11	2.73	0.05	0.65
+	-	**2150.00**	107.50	2.32	0.05	0.63
+	+	2804.28	**106.25**	**2.28**	0.05	0.63

5.1.2 Observations on Metrics

The experiments are conducted based on L_2 and LPIPS distance metrics. Since traditional L_p norm-based measures are based on pixel-based differences, they are not effective indicators for perceptual quality, hence, it is considered to use LPIPS perceptual distance as the primary metric in the evaluation. With regards to the proposed minimization methods, LPIPS-Minimization can be used in conjunction with any type of attack, while L_2-Minimization is less effective in some attack types such as the ones based on shifting of pixels (e.g., [37]). Nevertheless, the distances of L_2 and LPIPS minimization methods are measured with both L_2 and LPIPS metrics. It is seen that both distance metrics usually decrease with any of the Minimization Method. Though, as expected, the measured metric benefits more when it is the same metric as the one used in minimization (e.g., the LPIPS-minimization method minimizes LPIPS distance proportionally more compared to L_2 distance).

5.2 Alternative Methods for Normalized Variance Weighting

In this section, alternative methods for Normalized Variance Weighting are discussed. Primarily, the Variance-Based Box-Constrained Method is mentioned, then Quantile Gradient Norm Masking Method is explained. The advantages and limitations of both methods are discussed.

5.2.1 Variance-Based Box-Constrained Method

There are several previous works applying this approach to different types of attacks [30]. The main idea of this approach is to limit the maximum change in pixels so that perturbations should be imper-

Table 9: BIM attack results against ResNet50 with Quantile Gradient Norm Masking Method (Epsilon ϵ = 0.005, Quantile q = 0.7, alpha α = 0.00025, iteration = 20)

Grad. Mask	Minimization	L_0	L_1	L_2	L_∞	Fooling Rate
-	-	2939.46	11.85	0.23	0.01	0.45
-	+	3070.41	9.63	0.19	0.01	0.45
+	-	**2775.43**	9.79	0.20	0.01	0.43
+	+	3005.57	**7.86**	**0.16**	0.01	0.43

Table 10: BIM attack results against ResNet50 with Quantile Gradient Norm Masking Method (Epsilon ϵ = 0.05, Quantile q = 0.7, alpha α = 0.00025, iteration = 20)

Grad Mask	Minimization	L_0	L_1	L_2	L_∞	Fooling Rate
-	-	2578.33	56.31	1.26	0.05	1.00
-	+	3046.08	54.66	1.22	0.05	1.00
+	-	**2755.22**	49.98	1.12	0.05	1.00
+	+	3056.62	**48.09**	**1.08**	0.05	1.00

ceptible. We have tested this approach as an alternative to the proposed Normalized Variance Weighting Method in an attack-agnostic manner with a white-box setting. Variance-based box-constrained adversarial examples are improved in terms of perceptual quality as well, compared to regular FGSM attack. However, it is seen in Figure 7 that the Variance Weighting Method has lower LPIPS perceptual distances compared to the variance-based box-constrained method at all fooling rates compared to FGSM and BIM attacks on NIPS2017. The variance-based box-constrained method also requires an additional coefficient parameter for each adversarial attack type, dataset, and network. In Figure 7, the epsilon value is used as the additional coefficient parameter of the variance-based box-constrained method to select the adaptive coefficient. Even when parameters for threshold levels are optimized, in most instances, it is observed that variance-weighted perturbations have better perceptual quality, which makes Normalized Variance Weighting a better choice.

5.2.2 Quantile Gradient Norm Masking Method

This method is also developed as an alternative to the Normalized Variance Weighting Method as seen in Figure 5. The procedure of methods is highly similar. However, this method could only be applied to gradient-based white-box adversarial attack algorithms as it is shown in Table 7 to 11. Regarding that, this method is applicable to only gradient-based attacks, it is not proposed as a main method in the methodology chapter.

Primarily, gradients are calculated via any gradient-based adversarial algorithm. Then, the norm of gradients is sorted for all input variables (i.e., for all pixels). After acquiring the sorted norm of the gradients, a quantile mask is created in line with the given quantile as shown in Figure 6. This binarized mask removes the most ineffective input variables (i.e., 100 * (1 - q) percentage of the pixels). After this operation, the regular procedure of the existing adversarial attack algorithm will continue.

Table 11: CW attack results against ResNet50 with Quantile Gradient Norm Masking Method

Grad Mask	Minimization	L_0	L_1	L_2	L_∞	Fooling Rate
-	-	2965.84	11.97	0.28	0.02	0.97
-	+	2964.64	11.02	0.27	0.02	0.97
+	-	**1806.00**	8.08	0.23	0.02	0.84
+	+	**1805.99**	**7.49**	**0.22**	0.02	0.84

An example of Quantile Gradient Norm Masking for the FGSM [8] algorithm where q is set to 0.7 is as follows. Initially, the FGSM algorithm would compute the gradient. The Gradient Norm Masking Method will compute the quantile mask using the norm of the gradients. Then, the FGSM algorithm will compute the perturbation by multiplying the negative signs of the gradient with the epsilon coefficient. After the calculation of the perturbation, the perturbation will be masked with the generated quantile mask. As a result, the L_0 norm will decrease 100 * (1 - q) percentage.

CHAPTER 6

CONCLUSION AND FUTURE WORK

This study proposed an integrated method combined with two attack agnostic techniques to improve the perceptual quality of adversarial examples while preserving the fooling rate.

The Ultimate Method (i.e., integration of the variance weighting and minimization method) generates adversarial examples with the best perceptual quality measured by LPIPS. While the Normalized Variance Weighting Method would be applied during the attack, the Minimization Method could only be applied after the generation of successful adversarial examples. Regarding that, the integration of these two methods would become considerably effortless and the most effective method.

Applying the Variance Weighting Method improves the perceptual quality of different types of adversarial attacks without any extra significant computational cost in a white-box setting. If the system has time constraints (e.g., real-time CAPTCHA image generation), then FGSM [8] algorithm could be applied with the Normalized Variance Weighting Method effortlessly. Thus, the adversarial images could be generated for CAPTCHA while the perceptual quality of the examples is improved in a real-time setting.

Regarding the fact that the Minimization Method calculates gradient information at each iteration, this method is less suitable for real-time systems compared to the Normalized Variance Weighting Method. However, the Minimization Method reveals that adversarial examples which have been already generated could also be improved further. It is shown that perturbations produced by different types of adversarial attacks could be minimized while preserving the fooling rate. Thus, adversarial examples that are disturbing to human vision could be improved with respect to different types of distance metrics (i.e., L_2 or LPIPS).

As mentioned in Section 5.2, alternative methods (i.e., Variance-Based Box-Constrained Method and Quantile Gradient Norm Masking) could be further improved to build an effective technique. It is shown that in Figure 7 Variance-Based Box-Constrained Method is also improving the perceptual quality although not as effective as the Variance Weighting Method.

Quantile Gradient Norm Masking Method which is another alternative method to the Variance Weighting Method does not benefit from the variance information at all. It is shown in Tables 7, 8, 9, 10 and 11 that mask operation using gradient information could also be effective to improve perceptual quality. As seen in Figures 5 and 6, the histogram information could be effectively used. Several different methods could be produced to improve the perceptual quality of the adversarial examples using histogram information in various ways (e.g., minimum required epsilon for input-dependent adversarial attacks could be calculated using the histogram information).

Besides these techniques, the Normalized Variance Weighting Method could be improved further. As seen in Figure 4(d), applying the Normalized Variance Weighting Method may worsen the perceptual quality of images containing dominantly low-variance areas with only a small portion of high-variance areas. If images with this behavior could be detected, it would be better to avoid performing any operations on these detected images instead of applying the variance method on every image. Thus, an adaptive decision mechanism could be designed and inserted into the Normalized Variance Method. This will prevent worsening the perceptual quality of images with such behavior which will also reduce the computational cost of the variance method.

In addition, other attack agnostic improvements (e.g., generating adversarial attacks on YUV color space [6]) could be combined with proposed methods to enhance perceptual quality further in the future.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," *Neural Information Processing Systems*, vol. 25, 01 2012.

[2] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[3] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, 2014.

[4] D. Silver, A. Huang, C. Maddison, A. Guez, L. Sifre, G. Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484–489, 01 2016.

[5] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *International Conference on Learning Representations*, 2014.

[6] B. Aksoy and A. Temizel, "Attack type agnostic perceptual enhancement of adversarial images," in *International Workshop on Adversarial Machine Learning And Security (AMLAS), IEEE World Congress on Computational Intelligence (IEEE WCCI)*, 2019.

[7] C. Shi, X. Xu, S. Ji, K. Bu, J. Chen, R. A. Beyah, and T. Wang, "Adversarial captchas," *CoRR*, vol. abs/1901.01107, 2019.

[8] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *3rd International Conference on Learning Representations, ICLR* (Y. Bengio and Y. LeCun, eds.), 2015.

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

[10] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2574–2582, 2016.

[11] N. Papernot, P. D. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *IEEE European Symposium on Security and Privacy, EuroS&P 2016, Saarbrücken, Germany, March 21-24, 2016*, pp. 372–387, IEEE, 2016.

[12] C. Xiao, J. Y. Zhu, B. Li, W. He, M. Liu, and D. Song, "Spatially transformed adversarial examples," in *International Conference on Learning Representations (ICLR)*, 2018.

[13] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy)*, pp. 39–57, 2017.

[14] R. Zhang, P. Isola, A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595, 06 2018.

[15] B. T. Karli, D. Sen, and A. Temizel, "Improving perceptual quality of adversarial images using perceptual distance minimization and normalized variance weighting," in *The AAAI-22 Workshop on Adversarial Machine Learning and Beyond*, 2022.

[16] S. Qiu, Q. Liu, S. Zhou, and C. Wu, "Review of artificial intelligence adversarial attack and defense technologies," *Applied Sciences*, vol. 9, no. 5, 2019.

[17] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille, "Adversarial examples for semantic segmentation and object detection," in *International Conference on Computer Vision*, IEEE, 2017.

[18] B. Liang, H. Li, M. Su, P. Bian, X. Li, and W. Shi, "Deep text classification can be fooled," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, p. 4208–4215, AAAI Press, 2018.

[19] Y. Belinkov and Y. Bisk, "Synthetic and natural noise both break neural machine translation," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, OpenReview.net, 2018.

[20] N. Papernot, P. D. McDaniel, I. J. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, AsiaCCS 2017, Abu Dhabi, United Arab Emirates, April 2-6, 2017* (R. Karri, O. Sinanoglu, A. Sadeghi, and X. Yi, eds.), pp. 506–519, ACM, 2017.

[21] K. Grosse, N. Papernot, P. Manoharan, M. Backes, and P. D. McDaniel, "Adversarial perturbations against deep neural networks for malware classification," *CoRR*, vol. abs/1606.04435, 2016.

[22] C.-H. Huang, T.-H. Lee, L.-h. Chang, J.-R. Lin, and G. Horng, *Adversarial Attacks on SDN-Based Deep Learning IDS System: International Conference on Mobile and Wireless Technology (ICMWT 2018)*, pp. 181–191. 01 2019.

[23] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, OpenReview.net, 2017.

[24] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust Physical-World Attacks on Deep Learning Visual Classification," in *Computer Vision and Pattern Recognition (CVPR)*, 2018.

[25] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, (New York, NY, USA), p. 1528–1540, Association for Computing Machinery, 2016.

[26] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*, OpenReview.net, 2017.

[27] S. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," *CoRR*, vol. abs/1610.08401, 2016.

[28] D. Sen, B. T. Karli, and A. Temizel, "Training universal adversarial perturbations with alternating loss functions," in *The AAAI-22 Workshop on Adversarial Machine Learning and Beyond*, 2022.

[29] B. Luo, Y. Liu, L. Wei, and Q. Xu, "Towards imperceptible and robust adversarial example attacks against neural networks," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018* (S. A. McIlraith and K. Q. Weinberger, eds.), pp. 1652–1659, AAAI Press, 2018.

[30] F. Croce and M. Hein, "Sparse and imperceivable adversarial attacks," in *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pp. 4723–4731, IEEE, 2019.

[31] G. E. Legge and J. M. Foley, "Contrast masking in human vision," *J. Opt. Soc. Am.*, vol. 70, pp. 1458–1471, Dec 1980.

[32] W. Lin, L. Dong, and P. Xue, "Visual distortion gauge based on discrimination of noticeable contrast changes," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 7, pp. 900–909, 2005.

[33] A. Liu, W. Lin, M. Paul, C. Deng, and F. Zhang, "Just noticeable difference for images with decomposition model for separating edge and textured regions," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 11, pp. 1648–1652, 2010.

[34] M. Jordan, N. Manoj, S. Goel, and A. G. Dimakis, "Quantifying perceptual distortion of adversarial examples," *CoRR*, vol. abs/1902.08265, 2019.

[35] C. Laidlaw and S. Feizi, "Functional adversarial attacks," *Advances in Neural Information Processing Systems*, vol. 32, pp. 10408–10418, 2019.

[36] C. Laidlaw, S. Singla, and S. Feizi, "Perceptual adversarial robustness: Defense against unseen threat models," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, OpenReview.net, 2021.

[37] A. Aydin, D. Sen, B. T. Karli, O. Hanoglu, and A. Temizel, "Imperceptible adversarial examples by spatial chroma-shift," in *Proceedings of the 1st International Workshop on Adversarial Learning for Multimedia*, ADVM '21, (New York, NY, USA), p. 8–14, Association for Computing Machinery, 2021.

[38] K. Ding, K. Ma, S. Wang, and E. Simoncelli, "Image quality assessment: Unifying structure and texture similarity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, pp. 1–1, 12 2020.

[39] M. Sharif, L. Bauer, and M. Reiter, "On the suitability of lp-norms for creating and preventing adversarial examples," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1686–16868, 2018.

[40] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations (ICLR)*, 2015.

[41] 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 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 770–778, IEEE Computer Society, 2016.

[42] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016.

[43] N. Papernot, F. Faghri, N. Carlini, I. Goodfellow, R. Feinman, A. Kurakin, C. Xie, Y. Sharma, T. Brown, A. Roy, A. Matyasko, V. Behzadan, K. Hambardzumyan, Z. Zhang, Y.-L. Juang, Z. Li, R. Sheatsley, A. Garg, J. Uesato, W. Gierke, Y. Dong, D. Berthelot, P. Hendricks, J. Rauber, and R. Long, "Technical report on the cleverhans v2.1.0 adversarial examples library," *arXiv preprint arXiv:1610.00768*, 2018.

[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (Y. Bengio and Y. LeCun, eds.), 2015.