PERCEPTUAL QUALITY PRESERVING ADVERSARIAL ATTACKS

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

BY

BILGIN AKSOY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
MODELLING AND SIMULATION

SEPTEMBER 2019

Approval of the thesis:

PERCEPTUAL QUALITY PRESERVING ADVERSARIAL ATTACKS

submitted by **BILGIN AKSOY** in partial fulfillment of the requirements for the degree of **Master of Science in Modelling and Simulation Department, Middle East Technical University** by,

Prof. Dr. Deniz Zeyrek Bozşahin
Dean, Graduate School of **Informatics** _____

Assist. Prof. Dr. Elif Sürer
Head of Department, **Modelling and Simulation, METU** _____

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

Examining Committee Members:

Assoc. Prof. Dr. Banu Günel Kılıç
Information Systems Department, METU _____

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

Assoc. Prof. Dr. Hüseyin Hacıhabiboğlu
Modelling and Simulation Department, METU _____

Assist. Prof. Dr. Tolga İnan
Electrical and Electronics Engineering, Çankaya University _____

Assist. Prof. Dr. Hacer Yalım Keleş
Computer Engineering, Ankara University _____

Date: 02.09.2019

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: Bilgin Aksoy

Signature :

ABSTRACT

PERCEPTUAL QUALITY PRESERVING ADVERSARIAL ATTACKS

Aksoy, Bilgin
M.S., Department of Modelling and Simulation
Supervisor: Prof. Dr. Alptekin Temizel

September 2019, 33 pages

Deep learning is used in various succesful computer vision applications such as image classification. Deep neural networks (DNN) especially convolutional neural networks have reached above human level accuracy rates for image classification tasks. While DNNs have solved the image classification task and enabled its use in many practical applications, recent research has unveiled some properties which could degrade their performance. Adversarial images are samples that are intentionally modified by adding non-random noise to deceive deep learning systems. Even the-state-of-the-art networks fail classifying these adversarial images to the corresponding class. They are widely used in applications such as CAPTHAs to help distinguish legitimate human users from bots. However, the noise introduced during the adversarial image generation process degrades the perceptual quality and introduces artificial colors; making it also difficult for humans to classify images and recognize objects. This thesis proposes a method that enables generation of adversarial images while preserving their perceptual quality. The proposed method is attack type agnostic and could be used in association with the existing attacks in the literature. Experiments show that the generated adversarial images have lower Euclidean distances to their originals while maintaining the same adversarial attack performance. Distances are reduced by 0.0315% to 29.6% with an average reduction of 17.8% over the different attack and network types.

Keywords: deep learning, image classification, adversarial images, perceptual enhancement

ÖZ

ALGISAL KALİTE KORUNARAK ÇEKİŞMELİ ÖRNEK ÜRETİMİ

Aksoy, Bilgin
Yüksek Lisans, Modelleme ve Simülasyon Anabilim Dalı Bölümü
Tez Yöneticisi: Prof. Dr. Alptekin Temizel

Eylül 2019 , 33 sayfa

Derin öğrenmenin, imge sınıflandırma gibi bilgisayarlı görü görevlerinde birçok başarılı uygulaması bulunmaktadır. Derin öğrenme ağları imge sınıflandırma problemlerinde insan başarımının üstünde başarım elde etmektedir. Her ne kadar sonuçlar derin sinir ağlarının imge sınıflandırma görevini çözmüş olduğunu ve pek çok uygulamada kullanılmasını sağladığını gösterse de son zamanlarda derin sinir ağlarının bazı özelliklerinin performanslarının düşmesine sebep olabileceğini göstermiştir. Girdi imgeye rastgel olmayan gürültü eklenmesiyle elde edilen çekişmeli imge/örnek adıda verilen imgelerin en başarılı ağlar tarafından bile yanlış sınıflandırmasına neden olduğu gösterilmiştir. Bu CAPTCHA gibi uygulamalarda sıkça kullanılmaktadır. Ancak çekişmeli süreçte elde edilen örnekler renkli gürültüler nedeniyle algısal kalite olarak düşük olabilmekte ve insanlar tarafından tanınmada sorunlar olabilmektedir. Bu tez çekişmeli örneklerin algısal kalitesini iyileştiren bir metot önermektedir. Önerilen metot literatürdeki saldırı tiplerinden bağımsız iyileştirme sağlamaktadır. Yapılan deneyler göstermektedir ki üretilen çekişmeli örneklerin Öklid uzaklığı azaltılabilmekteyken aynı zamanda çekişmeli örneğin başarımı korunabilmektedir. L_2 mesafesi %0.0315 ile %29.6 arasında ortalama olarak da %17.8 oranında azaltmaktadır.

Anahtar Kelimeler: derin öğrenme, imge sınıflandırma, çekişmeli örnek, algısal iyileştirme

This thesis is dedicated to my family, my friends and the people that care about me.

ACKNOWLEDGEMENTS

Throughout preparing and writing this thesis, I have received great support and assistance from my advisor Prof.Dr. Alptekin Temizel. I would first like to thank him for his endless support. It has been a great journey with his support and assistance.

I would also like to thank to my wife, my daughter, my mother and my father without their support I wouldn't be able persevere throughout my journey.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

BIM	Basic Iterative Method
C&W L_2	Carlini&Wagner L_2
DNN	Deep Neural Networks
FGSM	Fast Gradient Sign Method
FR	Full-reference
HVS	Human Visual System
IncResV2	Inception Resnet V2 architecture
IncV3	Inception V3 architecture
IQA	Image Quality Assessment
JSMA	Jacobian-based Saliency Map Attack
L-BFGS	Limited Memory Broyden–Fletcher–Goldfarb–Shanno
MIM	Momentum Iterative Method
MSCN	Mean subtracted contrast normalized
NR	No-reference
PSNR	Peak Signal-to-Noise Ratio
Res50V3	ResNet50 V3 architecture
RGB	Red-Green-Blue color space
SSIM	Structural Similarity index
SVR	Support Vector Machines for Regression
YUV	YUV color space –Y stands for luminance component and U and V stands for chrominance components

CHAPTER 1

INTRODUCTION

Deep neural networks (DNNs) have been a popular choice in computer vision tasks since AlexNet[1] won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012 with a 11% better Top-5 error rate improvement in comprasion with the former winner in 2011. Since 2012, DNNs have dominated both ILSVRC and other visual challenges and adapted to other perceptual tasks.

1.1 Motivation and Problem Definition

An adversarial attack perturbs the input image by adding a non-random, network and input specific noise, to make its automated classification difficult. This artificial noise also makes it more difficult for the legitimate users to classify the adversarial images especially when they are time limited [2] as shown in Fig.1.1. Two desired attributes of adversarial images are:

- They should successfully fool the machine learning systems,
- They should introduce as little perceptual noise as possible so that they do not pose any additional challenge to the humans.

Completely **A**utomated **P**ublic **T**uring test to tell **C**omputers and **H**umans **A**part - CAPTCHA, is a commonly used method to validate human users. There are four commonly used CAPTCHA schemes[3]:

- Text-based CAPTCHA,
- Image-based CAPTCHA(see Fig.1.2),
- Audio-based CAPTCHA,
- Video-based CAPTCHA.

Image classification based tests are intentionally designed to make bots fail to classify images. This thesis mainly focuses on image CAPTCHA. Deep Neural Network (DNN) based methods [4, 5, 6], which have recently been proven to be successful in automated image classification, have been found to be useful to bypass CAPTCHA

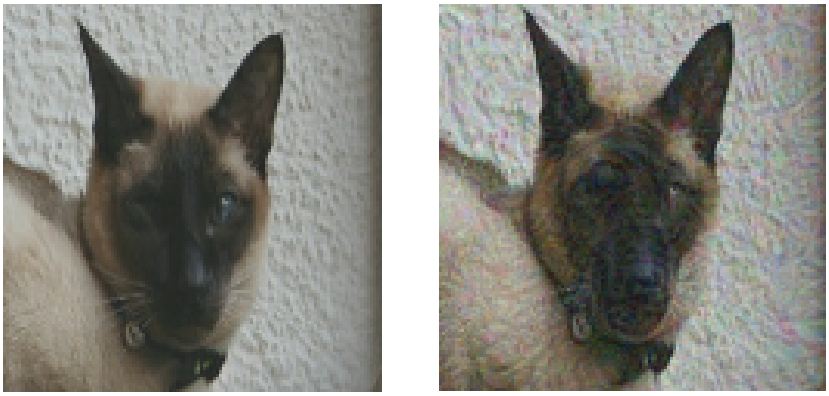


Figure 1.1: Original input and adversarial counterpart which is also difficult for humans in limited time. Adapted from 'Adversarial Examples that Fool both Computer Vision and Time-Limited Humans,' by G.F. Elsayed and S. Shankar and B. Cheung and N. Papernot and A. Kurakin and I. Goodfellow and J. Sohl-Dickstein. Reprinted with permisson.

security process. However, these methods are vulnerable to specially generated adversarial examples [7], which can be used in CAPTCHAs and similar applications to make them more robust.

Text-based and image-based CAPTCHA based on adversarial learning which is called aCAPTCHA was proposed in [3]. Since the performance of CAPTCHA is vital then it would likely be more disturbing for human.

1.2 Proposed Method and Model

This thesis proposes a method for perceptual enhancement of adversarial images to make them closer to their noise-free originals and easier to process by humans.

Since human visual system is more sensitive to luminance channel than chrominance channel, the proposed method seeks an adversarial image which has less noise in chrominance channels than luminance channel in order to make adversarial images more easy for human to recognize an object but preserve the bots would fail.

The proposed method first convert the adversarial noise to YUV color space and then reduce the noise in chrominance channel and finally apply Gaussian blurring kernel to those channels. The final result is more human recognizable but still having a similar level of performance.

1.3 Contributions and Novelties

The proposed method's contributions are as follows:

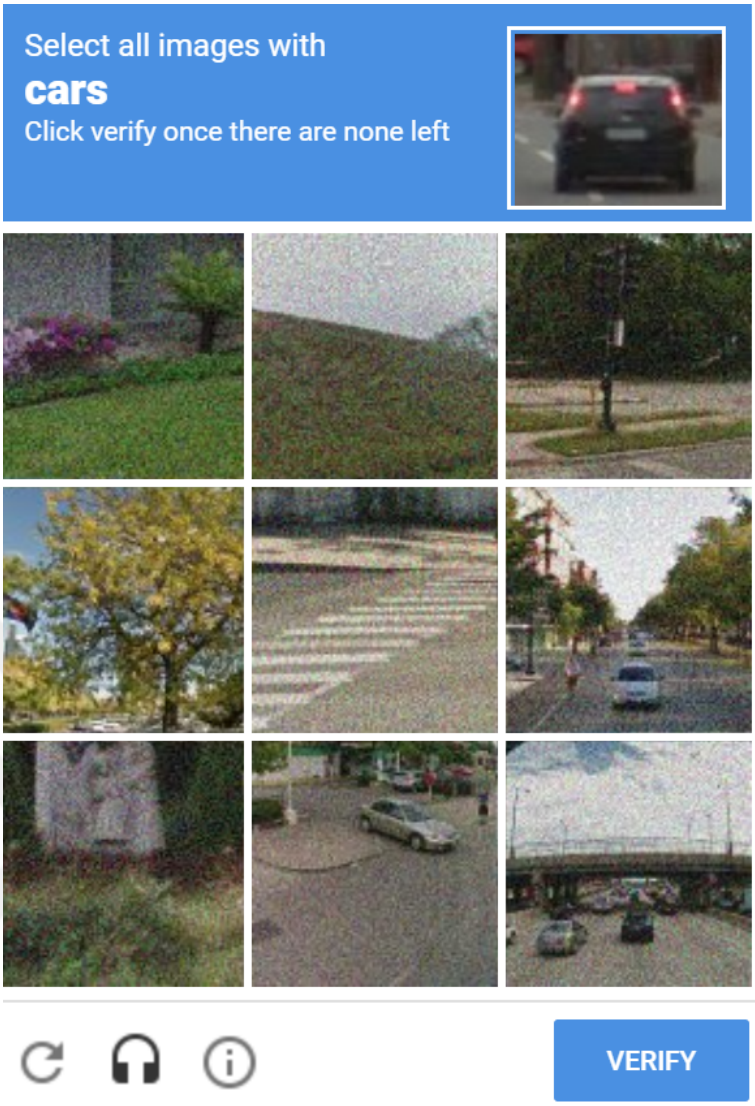


Figure 1.2: Google Image reCAPTCHA

- Reducing the distractive colored snow-like noise in adversarial image and the final result are more human recognizable,
- Preserving 100% the attack accuracy to fail bots,
- Lower noise levels can be obtained by concentrating the attack on the luminance channel.

1.4 The Outline of the Thesis

Chapter-2 introduces the adversarial attacks, distance metrics, RGB and YUV color spaces, Chapter-3 describes the methodology introduced in the thesis, and the algorithms, Chapter-4 presents the results obtained from proposed method, and Chapter-5

presents the conclusion.

CHAPTER 2

REVIEW OF LITERATURE

2.1 Introduction

An adversarial attack perturbs the input image by adding a non-random, network and input specific noise which is (ideally) hardly perceptible to make its automated classification difficult. This vulnerability was first introduced in [8]. Szegedy et al. [8] found that neural networks learn discontinuous input-output mappings. If the prediction error of the selected network is maximized, then there would be some imperceptible noise that can lead the network misclassify the given input. Meanwhile the adversarial image which is the summation of imperceptible noise and original image could be misclassified by a different network, that was trained on a different subset of the dataset.

2.2 Taxonomy of Adversarial Attacks

An adversarial attack generates some noise in order to cause deep neural networks misclassify the input. The computed noise is called perturbation or adversarial noise, and the final result can be computed adding the adversarial noise to the original input. Adversarial attacks can be taxonomically divided into subcategories according to the process of computing the adversarial noise, knowledge about target/defense network, and the the presence of the target label:

- Depending the generation of adversarial examples by means of usage iteration (iterative/non-iterative),
- Depending the generation of adversarial examples by means of knowledge about parameters of the network (white-box/black-box),
- Generating the adversarial examples to be classified as a specific class (targeted/non-targeted),
- Generating the adversarial noise for specific input or whole dataset (individual/universal perturbation)

Adversarial attacks often remained as an academic research topic. However, the use of adversarial attacks in bot systems is very new and Figure-1.2 and aCAPTCHA a

framework for text-based and image-based adversarial CAPTCHA generation [3] are early examples.

2.2.1 Iterative Attack vs Non-Iterative Attack

Iterative attack crafts the adversarial example in an iterative process which aims to find the minimum perturbation. *Non-iterative attack* crafts the perturbation after one process. Iterative attacks generally produces less perturbation than non-iterative attacks but as a side effect has higher computational complexity.

2.2.2 White Box vs Black Box Attack

Black box attack setup has no knowledge about the model, no access to the internal parameters and structure of the model[9] and has access only the output of the model (label or confidence score). But *white box attack* setup, the attack model has access to model's parameters and structure.

Most of the proposed adversarial attacks in the literature are white-box attacks. However, they can be converted to a black-box attack, due to the transferability of adversarial examples proposed by Papernot et al.[10]. Papernot et. al.[10] proposed that after training an adversarial process on a network, it is also possible to be misclassified the adversarial examples by another model which has completely different architecture, if the models had been trained on same task. Knowing the model parameters in real world isn't always possible, so black box models can be more efficient.

2.2.3 Targeted Attack vs Non-Targeted Attack

In a *targeted attack* setting, the aim is make an input being as classified as a sample from a specific class. For example, if the corresponding class of input is *Panda* and the target class after adversarial attack is *Gibbon*, the network is expected classify a Panda image as Gibbon. In a *Non-targeted attack* setup, no target class is specified and it aims to mislead the network regardless of the class label. In this case if the corresponding class of input is *Panda*, the network is expected to classify a Panda image as any class other than Panda.

2.2.4 Individual Perturbation vs Universal Perturbation

Individual perturbation is input specific. But *universal perturbation* can be applied to the whole dataset. But the adversarial image produced by adding individual perturbation can be more robust against defence models but as a side effect has higher computational complexity.

Most of the proposed adversarial attacks generate individual perturbation. However, universal perturbation is more applicable in real world then individual perturbation

[11].

2.3 Adversarial Attacks

Adversarial attacks have been a trending research topic in deep learning since Szegedy et al.[8] first introduced the intriguing properties of deep neural networks. The important one of these intriguing properties by context of adversarial examples is input-output mappings learned by neural networks are discontinuous and as a result adding non-random imperceptible noise can cause the network to misclassify the input by maximizing the network's prediction error. There are various adversarial attack method and will be explained throughout this section.

There are various various methods for adversarial image generation in literature. This thesis focuses on the attacks which are first introduced or considered as the most important breakthroughs in the area. The selected methods will be explained in detail throughout this section. L-BFGS, proposed by Szegedy et al. [7], is known to be the first adversarial attack method in the literature. Fast Gradient Sign Method (FGSM) [12] is an non-iterative attack. Basic Iterative Method [13] and Momentum Iterative Method (MIM) [14] are based on FGSM but different to FGSM they use multiple iterations to find the minimal perturbation. Carlini&Wagner (C&W) L_2 attack is an iterative attack which can be generalized to L_∞ norm. DeepFool [15] is also an iterative attack.

The distance between original input and adversarial counterpart is generally computed using full-reference metrics as these models are optimized using metrics such as L_p norm.

The main purpose of adversarial process is for a given input image $\boldsymbol{I}$ and corresponding label y finding an adversarial example $\boldsymbol{A}$ which could be classified by network function f as $\hat{y}$ where $\hat{y} \neq y$.

2.3.1 L-BFGS

$f : \mathbb{R}^m \longrightarrow \{1 \dots k\}$ is a classifier function which maps image $\boldsymbol{I}$ to a label y. For a given $\boldsymbol{I} \in \mathbb{R}^m$ image and label $y \in \{1 \dots k\}$ using penalty function:

$$\text{Minimize } c|\eta| + \text{loss}_f(\boldsymbol{I} + \eta, y) \text{ subject to } \boldsymbol{I} + \eta \in [0, 1]^m \tag{2.1}$$

to find the minimum $c > 0$ in iterative manner [8].

2.3.2 Fast Gradient Sign Method

Fast Gradient Sign Method (FGSM) [12] is a one-step gradient based approach which is designed to be fast. For a given image $\boldsymbol{I}$ and corresponding target y, it calculates the gradient of the loss, $\nabla_{\boldsymbol{I}} J(\boldsymbol{I}, y)$, generally cross-entropy, with respect to $\boldsymbol{I}$ and

multiplies negative of the gradient sign with a constant ϵ to generate the adversarial noise. This noise is then added to the image I to obtain the adversarial example $\boldsymbol{A}$ (2.2).

$$\boldsymbol{A} = \boldsymbol{I} + \epsilon \operatorname{sign}\left(\nabla_{\boldsymbol{I}} J(\boldsymbol{I}, y)\right) \tag{2.2}$$

2.3.3 Basic Iterative Method

Basic Iterative Method [13] is an iterative extension of FGSM which clips the pixel values in each iteration:

$$Clip_{\boldsymbol{I},\xi}\left\{\boldsymbol{I}'\right\} = \min\left\{255, \boldsymbol{I} + \xi, \max\left\{0, \boldsymbol{I} - \epsilon, \boldsymbol{I}'\right\}\right\} \tag{2.3}$$

and crafts the adversarial image A in multiple iterations:

$$\begin{aligned} \boldsymbol{I}_0 &= \boldsymbol{I} \\ \boldsymbol{I}_{n+1} &= Clip_{\boldsymbol{I},\xi}\left\{\boldsymbol{I}_n + \epsilon \operatorname{sign}\left(\nabla_{\boldsymbol{I}} J\left(\boldsymbol{I}_n, y\right)\right)\right\} \\ \boldsymbol{A} &= \boldsymbol{I}_{n+1} \end{aligned} \tag{2.4}$$

2.3.4 Momentum Iterative Method

Momentum Iterative Method (MIM) [14] is an iterative version of FGSM. It is designed to attain the minimum adversarial example in T iterations. At each iteration, MIM updates the accumulated gradient by using the current L_1 normalized gradient of loss, softmax cross-entropy, and previous accumulated gradient $\boldsymbol{g_t}$ multiplied by a decay factor μ (2.5). By this way, a momentum is used which makes the method more resilient to small humps, narrow valleys, and poor local extremities. Then the next adversarial example $\boldsymbol{A_{t+1}}$ is obtained by subtracting L_2 normalized $\boldsymbol{g_{t+1}}$ multiplied with a constant $\beta = \frac{\epsilon}{T}$.

$$\boldsymbol{g}_{t+1} = \mu \cdot \boldsymbol{g}_t + \frac{\nabla_{\boldsymbol{I}} J\left(\boldsymbol{A}_t, y\right)}{\left\|\nabla_{\boldsymbol{I}} J\left(\boldsymbol{A}_t, y\right)\right\|_1} \tag{2.5}$$

where $\boldsymbol{g}_t = 0$ when $t = 0$,

$$\boldsymbol{A}_{t+1} = \boldsymbol{A}_t - \beta \cdot \frac{\boldsymbol{g}_{t+1}}{\left\|\boldsymbol{g}_{t+1}\right\|_2} \tag{2.6}$$

2.3.5 Carlini&Wagner L_2

Carlini&Wagner (C&W) L_2 attack [16] aims to find the lowest perturbation in L_2 distance metric, also in an iterative manner. At each iteration, the attack finds the perturbation w for a given input image $\boldsymbol{I}$ and target class t by solving (2.7)

$$\text{minimize} \left\|\frac{1}{2}(\tanh(w) + 1) - \boldsymbol{I}\right\|_2^2 + c \cdot f\left(\frac{1}{2}(\tanh(w) + 1)\right) \tag{2.7}$$

where c is a positive constant and f is defined as in (2.8)

$$f\left(\boldsymbol{A}\right) = \max\left(\max\left\{Z\left(\boldsymbol{A}\right)_i : i \neq t\right\} - Z\left(\boldsymbol{A}\right)_t, -\kappa\right) \tag{2.8}$$

where Z is the activation function and κ is the confidence parameter, (how confident the classifier should be that the generated adversarial image is a sample of the target class). This thesis uses a non-targeted setup so that t is any incorrect class.

2.3.6 DeepFool

DeepFool [15] aims to find the minimum perturbation from the original input to the decision boundary of adversarial examples in an iterative manner. DeepFool first calculates the perturbation:

$$\eta^*(\boldsymbol{I}) = -\frac{f(\boldsymbol{I})}{\|w\|^2} w \tag{2.9}$$

of an affine classifier f. For a binary classifier the minimal perturbation is computed as:

$$\begin{aligned} &\arg\min_{\eta_i} && \|\eta_i\|_2 \\ &\text{Subjected to} && f\left(\boldsymbol{I}_i\right) + \nabla f\left(\boldsymbol{I}_i\right)^T \eta_i = 0 \end{aligned} \tag{2.10}$$

DeepFool can also be extended to multi-class classifier by finding the closest hyper-planes.

2.4 Human Visual System and Color Spaces

Human visual system (HVS) has two connected parts: the eye and the brain[17]. The eye's task in HVS is acquiring the radiating energy (light) reflected from a scene and sending it to the brain. HVS captures the information relating to the wavelength of light that different chromatic stimuli in any scene. The different chromatic stimuli in a scene constructs colors. So, color has trivariant coming from chromatic stimuli: hue, brightness and saturation and it is totally perceptual[18].

Human eye has cones, long-medium-short, which are responsible for detecting red, green, and blue colors. After detection, the next phase is discrimination of colors. Long and medium cones compute the red versus green color. Long, medium, and short cones together computes the blue versus green color. RGB color scheme is also inspiring from the human eye but it is not totally identical. In RGB color space, color information is separated into three different channels –red, green, and blue. But human eye is more sensitive to lightness information than color information in order to recognize the content. Some color spaces like YUV, YCbCr etc. focus this information. YUV color space has three channels –Y (luminance channel), U and V (chrominance channels). YUV and other derivatives are used generally in digital broadcasting. There are various color spaces. Selecting appropriate color space depends on the purpose. Equation-2.11 shows the conversion from RGB color

space to YUV color space, and Equation-2.12 shows the conversion from YUV color space to RGB color space.

$$\begin{aligned} Y &= (0.257 * R) + (0.504 * G) + (0.098 * B) + 16 \\ U &= -(0.148 * R) - (0.291 * G) + (0.439 * B) + 128 \\ V &= (0.439 * R) - (0.368 * G) - (0.071 * B) + 128 \end{aligned} \tag{2.11}$$

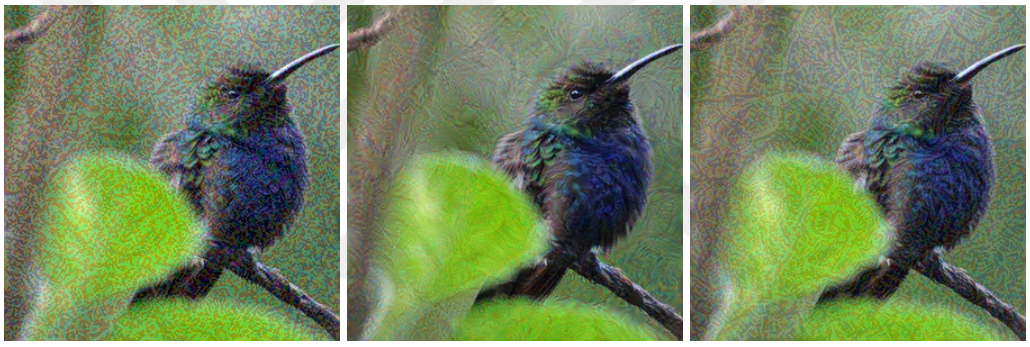
$$\begin{aligned} R &= 1.164(Y - 16) + 1.596(V - 128) \\ G &= 1.164(Y - 16) - 0.813(V - 128) - 0.391(U - 128) \\ B &= 1.164(Y - 16) + 2.018(U - 128) \end{aligned} \tag{2.12}$$

Separation of the luminance (lightness) information from the color information is better for human eye [19], since the human eye is more sensitive to lightness information of the scene. The proposed method utilizes human visual perception's basis in order to enhance adversarial images perceptually. The proposed method scales the noise in chrominance (U and V channels in YUV color space) channel as explained in Chapter 3.

CHAPTER 3

METHODOLOGY

The inputs of conventional DNNs are RGB images and all known attacks add noise to all three channels separately. Attack algorithms calculate the adversarial noise either using the network loss or the gradient of the network loss or the features extracted from saliency map. So the resulting adversarial noise has components in all three channels of RGB color space. Adding independent and different amounts of noise to these different channels results in artificial colors being introduced as shown in Fig.3.1a, 3.1b, 3.1c.



(a) **(b)** **(c)**

Figure 3.1: Articial colors in sample adversarial images obtained using different attacks (a) Adversarial image (FGSM attack), (b) Adversarial image (C&W L_2 attack), (c) Adversarial image (MIM attack)

Since the adversarial noise is a result of network internal parameters, the amount of noise can be controlled. Attack algorithms have been parameterized to adjust the adversarial noise. If the amount of adversarial noise is too small then the defense network cannot be failed. On the contrary, if the amount of adversarial noise is too large then it becomes difficult for humans to perceive the contents of the image.

In addition, as the attack modifies each pixel independently, it exhibits itself as a visually distractive colored snow-like high-frequency noise [20]. On the other hand, main distinguishing features (such as shape and texture) for an object class can be obtained from the luminance and adversarial noise added to the luminance channel is expected to be more detrimental to the network performance than the noise in the color channels. So, this thesis claims that lower noise levels could be obtained by

concentrating the attack on the luminance channel, which in effect is expected to reduce the distractive colored snow-like noise and the final result would be better in terms of human visual perception.

As conventional networks work with RGB images, the adversarial noise calculation inherently makes use of R, G and B channels. For the original image $I^{R,G,B}$, attack algorithm calculates the adversarial noise, $N^{R,G,B}$, separately for each channel. This noise is then added to the respective channels of the original image to obtain adversarial image $A^{R,G,B}$ as follows:

$$A^{R,G,B} = I^{R,G,B} + N^{R,G,B}. \tag{3.1}$$

The attack algorithms use either the network loss or the features extracted from saliency map to calculate the adversarial noise. Since the adversarial noise calculated by those coarse features which have little knowledge about the edge, corners, neighborhood of pixels etc., the calculated noise can be discomfortable for humans.

The proposed method first converts the image and the adversarial noise into YUV domain and obtain $I^{Y,U,V}$ and $N^{Y,U,V}$ respectively. Then U and V coefficients of the noise, N^U and N^V, are scaled by a factor $0 \leq \alpha \leq 1$. Assuming that the target object is closer to the center of the image, all the noise channels $N^{Y,U,V}$ are filtered with a 2D Gaussian kernel placed at the center of the image to gradually reduce the noise closer to the edges. The resulting noise is added in YUV color space:

$$A^{Y,U,V} = I^{Y,U,V} + N^{Y,U,V}. \tag{3.2}$$

Then the image $A^{Y,U,V}$ is converted back into RGB to allow processing in conventional networks. This process reduces the total amount of noise added to the original image and it might cause the adversarial attack to fail. Hence an iterative process is used as described in Alg.1 to find a stronger attack. Although a stronger attack will increase the noise, overall noise is lower due to the subsequent scaling of chrominance values and the use of Gaussian kernel.

3.1 Dataset

NIPS 2017: Adversarial Learning Development Set [21] consist of 1000 images having 299x299 resolution. Each image corresponds to a different ImageNet 1000 category. Image pixels are scaled to the range $[0, 1]$. All the images are used in the experiments and overall distance metrics are calculated as the average throughout all the images. 900 images of the dataset used for finding the best α and 100 of them used tor testing.

3.2 Selected Attacks

FGSM [12], BIM [22], MIM [14], C&W L_2 [16], and DeepFool [15] attacks were used for experimental evaluation of the proposed method as they are well-known milestone attacks.

Algorithm 1 Iteratively Finding the Minimum Adversarial Noise

Convert the original image $I^{R,G,B}$ into YUV: $I^{Y,U,V}$
Initialize the best distance L_2' to a high number
while Attack is successful **do**
    Run the attack to generate adversarial noise image $N^{R,G,B}$
    Convert $N^{R,G,B}$ into YUV: $N^{Y,U,V}$
    Scale the noise in U and V channels by a factor of α, apply Gaussian smoothing G to all noise channels and construct the adversarial image:
    $A^Y = I^Y + G(N^Y)$
    $A^U = I^U + G(\alpha \times N^U)$
    $A^V = I^V + G(\alpha \times N^V)$
    Convert $A^{Y,U,V}$ into RGB: $A^{R,G,B}$
    Calculate the new distance L_2 using $A^{R,G,B}$ and $I^{R,G,B}$
    if $L_2 < L_2'$ and attack is successful **then**
        Store the best attack:
        $A'^{R,G,B} = A^{R,G,B}$
        Store the minimum L_2 value as the new minimum
        $L_2' = L_2$
        Decrease the attack strength (ϵ for FGSM, MIM and BIM, maximum iteration for C&W L_2 and DeepFool)
    else return $A'^{R,G,B}$
    end if
end while

3.3 Experiment Setup and Pretrained Networks

3.3.1 Attacks' Implementation

Cleverhans module [23] was used for implementing the attacks. Each attack was trained in an untargeted setup and defended on three different well-known network architectures which are pretrained on ImageNet: Inception v3 (IncV3) [24], InceptionResNet v2 (IncResV2) [25], and ResNet50 v3 (Res50V3) [26].

There are two networks in an adversarial generation process. The first is an attack network which is used for generating the adversarial image, and the second is a defense network which is used for classifying the final adversarial image. The adversarial and defense networks are selected as same network in order to avoid attack performance loss.

3.3.2 Experiment Setup

The experiments aim that all attacks are successful, i.e., is the adversarial image generated by the attack network is misclassified by the defense network. To this end, ϵ parameter is used for FGSM, MIM, and BIM attacks and iteration parameter is used for C&W L_2 to find the minimum L_2 making the attack successful for each image. The images are downscaled to 224x224 for Res50v3 and they are kept at their origi-

nal resolution (229x299) for IncV3 and IncresV2. For all attack types, the Gaussian kernel size is set to match the size of the image and it has a standard deviation of 190 founded after a binary search algorithm.

3.3.3 Selection of Attacks' Parameters

FGSM attack's main parameter is ϵ which controls the attack step size and total perturbation. ϵ parameter is selected as 0.04 at the first iteration and decreased by 0.002 until the minimum ϵ which makes the defense network misclassify the adversarial image is obtained. If the adversarial attack fails at the first iteration then ϵ is increased by 0.04 and if it is successful then decreased by 0.002 until the minimum ϵ that makes the network misclassify the input is obtained.

For MIM attack, ϵ parameter is selected as 0.0018 for the first iteration and decreased by 0.001 until the minimum L_2 distance is obtained. Since MIM attack is based on FGSM attack, ϵ parameter controls the attack step size and total perturbation.

BIM attack ϵ parameter is selected as 0.0018 for the first iteration and decreased by 0.001 until the minimum L_2 distance is obtained.

C&W L_2 attack is initialized by setting confidence parameter to zero. Then the iteration parameter is increased, as long as the attack is successful, to find the minimum L_2 distance. Confidence parameter's higher values produces examples with larger L_2 distortion, but more strongly classified as adversarial. The iteration parameter's larger values produces lower distortion results.

DeepFool attack is initialized by setting maximum iteration 500. After each successful attack maximum iteration is increased by 100. Because maximum iteration parameter effects total perturbation.

3.4 Evaluation Metrics

Image Quality Assesment (IQA) process can be separated into two parts: Full-reference image quality assessment (FR-IQA) and no-reference image quality assessment (NR-IQA). FR-IQA process has information about the original input which is not perturbed. But NR-IQA process has no information about the original input.

Using FR-IQA process is a common practice in adversarial attacks since the adversarial optimization schemes are generally using the distance between original input and it's adversarial counterpart. All known adversarial research outputs evaluate their results using at least one of L_0, L_2, and L_∞ distance metrics.

3.4.1 Full-Reference Image Quality Assessment

Full-reference (FR) algorithms requires as input not only the perturbed image, but also a reference image which is not degraded. Since the original input image is known

before the adversarial process, FR-IQA metrics is a common practice in calculating the distance between input and output.

3.4.1.1 Error-Based Metrics

L_0, L_2, and L_∞ distances are commonly measures to quantify the perturbation added to the original image.

- L_0 distance counts the number of pixels which were altered during the adversarial process.
- L_∞ distance shows the maximum change due to the perturbation.
- Since the proposed method in this thesis aims perceptual enhancement, L_2 metric is used to calculate the total perturbation using all the channels (3.3).

$$L_2 = \sqrt{\sum_{c}^{R,G,B} \sum_{i=0}^{w} \sum_{j=0}^{h} (\boldsymbol{I}_{i,j}^{c} - \boldsymbol{A}_{i,j}^{c})^2} \tag{3.3}$$

  In this equation, $\boldsymbol{I}$ is the original image, $\boldsymbol{A}$ is the adversarial image, w is the width and, h is the height of the image. L_2 distance gives a better indication of the overall adversarial noise (high frequency noise which is distractive to human visual system) compared to L_0 and L_∞.

3.4.1.2 Perception Based Metrics

Perception-based distance metrics are used to calculate structural information degradation. Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM) [27] index are used commonly to calculate the perceptual distance between an original image and a degraded image. The degraded image should be considered as the adversarial image in the context of this thesis.

- PSNR is an expression for the ratio between the maximum possible value (power) of a signal and the power of distorting noise that affects the quality of its representation.

$$PSNR = 20 \log_{10} \left(\frac{MAX_{\boldsymbol{I}}}{\sqrt{MSE}} \right) \tag{3.4}$$

  where MSE is equal to L_2 distance Equation 3.3.

- SSIM measures the the perceptual difference between the original image and the degraded image. Again the degraded image should be considered as the adversarial image in the context of this thesis.

$$\mathrm{SSIM}(\mathbf{x}, \mathbf{y}) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{3.5}$$

where μ_x and μ_y are means of original image and degraded image respectively, σ_x and σ_y are standard deviation original image and degraded image respectively. C_1 is luminance comparison and C_2 is chrominance comparison, they are calculated respectively as:

$$\begin{aligned} C_1 &= (K_1 L)^2 \\ C_2 &= (K_2 L)^2 \end{aligned} \tag{3.6}$$

and K_1 and K_2 is a small constant and $K_{1,2} << 1$. SSIM index is selected, because SSIM is better performing for calculating the perceptual distance [27].

3.4.2 No-Reference Image Quality Assessment

No-reference (NR) image quality assessment (IQA) algorithms requires the only information about the perturbed image whose quality is being assessed[28]. NR-IQA algorithms extracts the statistical properties from a given dataset which consists generally natural images.

- *BLIINDS-II* (BLind Image Integrity Notator using DCT Statistics) algorithm [29] computes natural scene statistics (NSS) based 2-dimensional local DCT coefficients for $n \times n$ patches from image for different scales in first stage. The second stage of BLIINDS-II algorithm fits a generalized Gaussian model to DCT coefficients of local patches. The third stage extracts features from local patches for 3 orientations due to the fact that perturbation generally modify local orientation energy. The final stage in algorithm is a Bayesian model to predict the score. The whole process can be seen in Fig.3.2

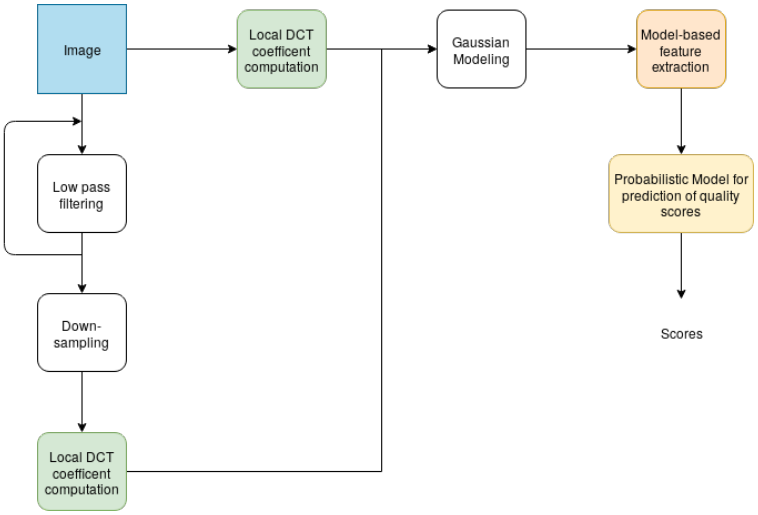


Figure 3.2: BLIINDS-2 Algorithm

- *NIQE* (Natural Image Quality Evaluator) algorithm [30] preprocesses the input by removing mean and applies normalization. It has observed that computed

coefficients 3.7 follows a Gaussian distribution for natural clean images. The second stage consists applying a Multivariate Gaussian Model (MVG) to the perturbed image features. The final stage is calculating the distance between NSS features and MVG fitted perturbed image features. The whole process can be seen in Fig.3.3

$$\hat{\boldsymbol{I}}(i,j) = \frac{\boldsymbol{I}(i,j) - \mu(i,j)}{\sigma(i,j) + 1} \tag{3.7}$$

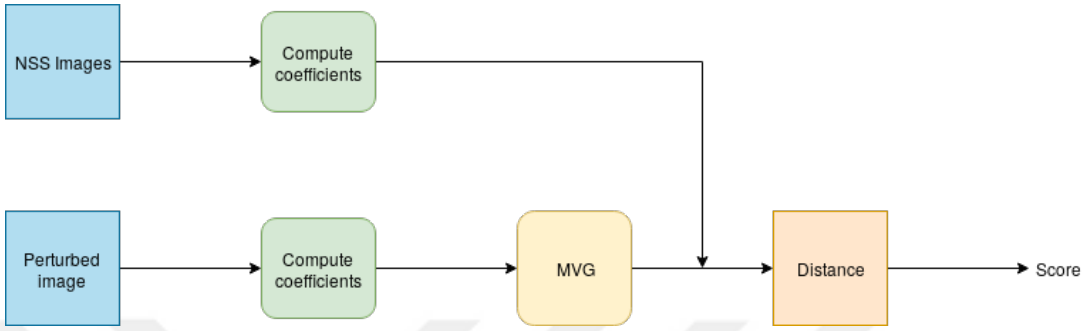


Figure 3.3: NIQE Algorithm

- *BRISQUE* (Blind/Referenceless Image Spatial Quality Evaluator) algorithm [31] is a blind/referenceless image quality assessment algorithm. Mean subtracted contrast normalized (MSCN) coefficients are computed in first stage using Eq.3.8. It has observed that MSCN coefficients for clean images fit generalized Gaussian distribution. Quality evaluation is computed using Support Vector Machines for regression (SVR).

$$\hat{\boldsymbol{I}}(i,j) = \frac{\boldsymbol{I}(i,j) - \mu(i,j)}{\sigma(i,j) + C} \tag{3.8}$$

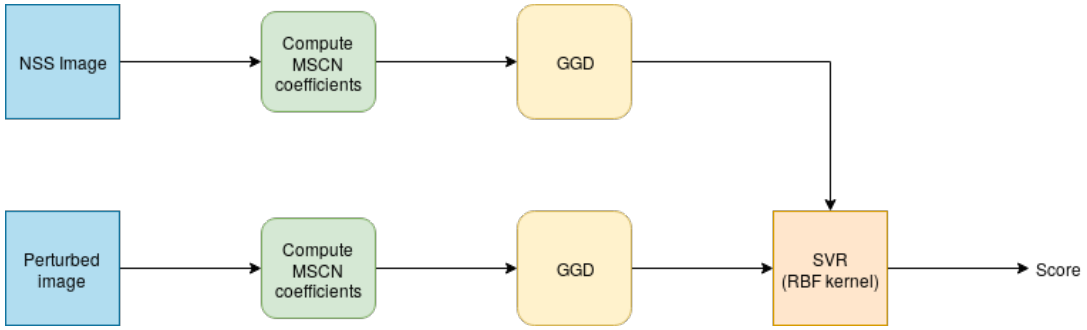


Figure 3.4: BRISQUE Algorithm

3.4.3 Tests

The proposed method finds the minimum distortion in an iterative manner which increases the time complexity. After finding the best α, all attacks have tested for

their best parameters as described in Alg.2 for new images which have not used for finding the best α.

Algorithm 2 Using the Best α

Convert the original image $I^{R,G,B}$ into YUV: $I^{Y,U,V}$
Run the attack to generate adversarial noise image $N^{R,G,B}$ for best attack parameters
Convert $N^{R,G,B}$ into YUV: $N^{Y,U,V}$
Scale the noise in U and V channels by using best α, apply Gaussian smoothing G to all noise channels and construct the adversarial image:
$A^Y = I^Y + G(N^Y)$
$A^U = I^U + G(\alpha \times N^U)$
$A^V = I^V + G(\alpha \times N^V)$
Convert $A^{Y,U,V}$ into RGB: $A^{R,G,B}$
return $A^{R,G,B}$

CHAPTER 4

RESULTS

4.1 Finding Best α

All results are given in Tables 4.1 and 4.2 for different values showing the mean value and standard deviation (removed outliers) of L2 distances respectively. Note that the case where α is 1 still has an effect of reducing the noise due to the Gaussian smoothing. When α is 0, no noise is added to the color channels. According to these results, for different attack and network types, there are different best α values:

- *FGSM*– The best α value is 0.2 for all attack networks.
- *C&W* L_2– The best α value is (0.2, 0.4, 0.6) for IncV3 and IncResV2 0.6 for Res50V3.
- *MIM*– The best α value is found as 0.2 for all attack networks.
- *BIM*– The best α value is 0.2 for all attack networks.
- *DeepFool*– The best α value is (0.2, 0.4, 0.6) for IncV3, (0.4, 0.6) for IncResV2 and 0.6 for Res50V3.

4.2 L_2 Distance Improvements

Since adversarial attacks generate different adversarial image for each attack network, L_2 distance improvement has been resulted different for each attach types and adversarial networks.

L_2 distance improvements on train set are computed as below:

- *FGSM*– The proposed method reduces the L_2 distance by 26% using IncV3 and IncResV2 pretrained model, and by 16.5% using Res50V3 pretrained model.
- *C&W* L_2– The proposed method reduces the L_2 distance by 19.1% using IncV3 pretrained model, by 0.078% using IncResV2 pretrained model, and by 0.0315% using Res50V3 pretrained model.
- *MIM*– The proposed method reduces the L_2 distance for MIM attack by 29% using IncV3 pretrained model, by 25.7% using IncResV2 pretrained model, and by 17.1% using Res50V3 pretrained model.

Table 4.1: L_2 distances for different attacks and different networks using various α values

Method	FGSM			C&W L_2			MIM			BIM			DeepFool		
	IncV3	IncResV2	Res50V3	IncV3	IncResV2	Res50V3	IncV3	IncResV2	Res50V3	IncV3	IncResV2	Res50V3	IncV3	IncResV2	Res50V3
Baseline	1.79	1.46	0.74	0.22	0.36	7.48	0.84	0.86	0.67	3.41	3.68	5.43	0.19	0.73	12.92
$\alpha = 1$	1.61	1.29	0.68	0.21	0.34	7.19	0.77	0.77	0.62	2.91	3.26	4.96	0.18	0.64	12.16
$\alpha = 0.8$	1.51	1.22	0.66	0.19	0.34	7.14	0.70	0.72	0.59	2.84	3.07	4.82	0.17	0.62	**12.08**
$\alpha = 0.6$	1.38	1.11	0.64	**0.18**	**0.33**	**7.13**	0.65	0.68	0.57	2.64	2.94	4.72	**0.16**	**0.61**	12.09
$\alpha = 0.4$	1.37	**1.11**	0.62	**0.18**	**0.33**	7.30	0.61	0.66	0.56	2.50	2.85	4.66	**0.16**	**0.61**	12.18
$\alpha = 0.2$	**1.33**	**1.11**	**0.61**	**0.18**	**0.33**	7.30	**0.59**	**0.65**	**0.55**	**2.45**	**2.82**	**4.63**	**0.16**	0.62	12.36
$\alpha = 0$	1.35	1.12	0.62	0.19	0.34	7.47	0.60	**0.65**	0.56	2.48	2.84	4.65	0.17	0.63	12.62

Table 4.2: Standard deviation of L_2 distances

Method	FGSM			C&W L_2			MIM			BIM			DeepFool		
	IncV3	IncResV2	Res50V3	IncV3	IncResV2	Res50V3	IncV3	IncResV2	Res50V3	IncV3	IncResV2	Res50V3	IncV3	IncResV2	Res50V3
$\alpha = 1$	0.011	0.099	0.009	0.164	0.167	0.161	0.01	0.01	0.008	0.014	0.016	0.014	0.127	0.672	3.530
$\alpha = 0.8$	0.088	0.129	0.076	0.157	0.161	0.157	0.01	0.01	0.007	0.011	0.015	0.012	0.121	0.657	3.497
$\alpha = 0.6$	0.075	0.081	0.074	0.152	0.16	0.153	0.013	0.012	0.007	0.009	0.013	0.012	0.116	0.650	3.490
$\alpha = 0.4$	0.076	0.081	0.072	0.15	0.153	0.129	0.016	0.014	0.008	0.008	0.011	0.008	0.114	0.65	3.508
$\alpha = 0.2$	0.075	0.188	0.071	0.148	0.151	0.122	0.019	0.015	0.008	0.007	0.009	0.007	0.115	0.658	3.553
$\alpha = 0$	0.076	0.189	0.071	0.153	0.154	0.131	0.019	0.015	0.008	0.008	0.012	0.008	0.118	0.673	3.621

- *BIM*– The proposed method reduces the L_2 distance by 29.6% using IncV3 pretrained model, by 24.3% using IncResV2 pretrained model, and by 14.6% using Res50V3 pretrained model.
- *DeepFool*– The proposed method reduces the L_2 distance by 19.4% using IncV3 pretrained model, by 18.8% using IncResV2 pretrained model, and by 0.055% using Res50V3 pretrained model.

4.3 Perceptual Evaluation of Result

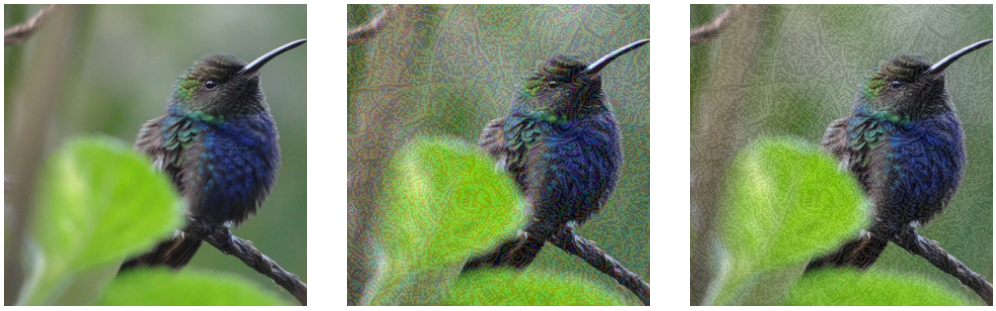
Figs. 4.1-4.5 shows baseline adversarial images and the images obtained with the proposed method for FGSM, C&W L_2, MIM, BIM, and DeepFool attacks. The result of proposed method is better not only when evaluating distance metrics but perceptually looking. The proposed method removes colorful noise and some fluctuations. Since DeepFool attack is well optimized attack, perceptual and adversarial results look alike.

(a) **(b)** **(c)**

Figure 4.1: (a) Original image, (b) Baseline adversarial image (FGSM attack), (c) Adversarial image obtained with $\alpha = 0$ (FGSM attack)

(a) **(b)** **(c)**

Figure 4.2: (a) Original image, (b) Baseline adversarial image (C&W L_2 attack), (c) Adversarial image obtained with $\alpha = 0$ (C&W L_2 attack)



(a) **(b)** **(c)**

Figure 4.3: (a) Original image, (b) Baseline adversarial image (MIM attack), (c) Adversarial image obtained with $\alpha = 0$ (MIM attack)

(a) **(b)** **(c)**

Figure 4.4: (a) Original image, (b) Baseline adversarial image (BIM attack), (c) Adversarial image obtained with $\alpha = 0$ (BIM attack)

(a) **(b)** **(c)**

Figure 4.5: (a) Original image, (b) Baseline adversarial image (DeepFool attack), (c) Adversarial image obtained with $\alpha = 0$ (DeepFool attack)

4.4 Test

The proposed method workflow finds the best α value for each attack type and network in an iterative manner. So, the workflow is very time consuming. Alg.2 directly uses the best α values found by using Alg.1 to speed-up the workflow in real world usage. Table-4.3 show that using Alg.2 results in a little (1-2%) attack accuracy loss and no iteration is needed.

Table 4.3: Attack Accuracy for Test Images Using Alg.2

	FGSM	C&W L_2	MIM	BIM	DeepFool
IncV3	100	100	98	99	100
IncResV2	100	100	98	97	100
Res50V2	100	100	100	100	100

Non-reference image quality assessment explained in Sec.3.4.2 in adversarial generation has not been reported in current the-state-of-the-art research in the literature. NIQE [30], BRISQUE [31], and BLIINDS-II [29] scores have been computed for test images. While the lower score for NIQE and BRISQUE indicates less perturbed image, the higher score for BLIINDS-II indicates less perturbed image. The proposed method has achieved better results for NIQE and BRISQUE algorithms. But the proposed method has been beaten by the original attacks for BLIINDS-II algorithm. The main difference between NIQE/BRISQUE algorithm and BLIINDS-II algorithm is BLIINDS-II extract features in frequency domain not in spatial domain. the scores obtained by BLIINDS-II algorithm don't always correspond to human judgment [35]. However the proposed method have beaten by baseline attacks judging the BLIINDS-II scores, still the resulting images are better in terms of human visual perception.

Table 4.4: NIQE Scores for Test Images

	FGSM		C&W L_2		MIM		BIM		DeepFool	
	Adv	Per	Adv	Per	Adv	Per	Adv	Per	Adv	Per
IncV3	4.34	**4.3**	4.25	**4.25**	**4.68**	4.69	11.65	**10.36**	**4.36**	4.39
IncResV2	4.25	**4.22**	4.40	**4.38**	**4.68**	4.68	11.62	**10.41**	**4.40**	4.41
Res50V3	5.32	**5.15**	**5.73**	5.74	5.17	**5.17**	25.87	**22.90**	5.7673	**5.66**

Table 4.5: BRISQUE Scores for Test Images

	FGSM		C&W L_2		MIM		BIM		DeepFool	
	Adv	Per	Adv	Per	Adv	Per	Adv	Per	Adv	Per
IncV3	25.61	**25.37**	26.49	**25.60**	**26.50**	26.72	44.70	**44.11**	**26.61**	26.71
IncResV2	25.49	**25.05**	26.45	**25.58**	**26.10**	26.33	44.92	**44.39**	**25.46**	25.57
Res50V3	27.34	**27.05**	27.95	**27.15**	**27.73**	27.85	44.24	**43.95**	23.91	**23.67**

Table 4.6: BLIINDS-II Scores for Test Images

	FGSM		C&W L_2		MIM		BIM		DeepFool	
	Adv	Per	Adv	Per	Adv	Per	Adv	Per	Adv	Per
IncV3	**12.05**	11.30	**13.02**	11.13	**15.19**	14.73	**36.80**	9.71	**15.36**	15.33
IncResV2	11.18	**11.44**	**12.54**	11.11	14.45	**14.63**	**34.47**	7.74	**14.16**	14.06
Res50V3	12.56	**13.12**	**13.91**	12.62	16.36	**16.45**	**26.55**	6.06	**10.58**	10.16

4.5 Discussion

Reducing the noise in U and V bands makes the adversarial images look perceptually better. However, in order to achieve 100% attack accuracy, stronger attacks, which increase the noise in Y, are needed as a trade-off. However, as can be seen in Table 4.1, lower or middle L_2 distances can still be obtained for all attack types and for all networks. It has to be noted that the α value giving the best result is different for each attack.

The results show that the proposed method works independent of the attack type and the network model and reduces the L_2 distances and better NIQE, and BRISQUE scores. Even though C&W L_2, BIM, DeepFool and MIM attacks are optimized to minimize L_2 distance by design, proposed method results in still lower L_2 values. While this might sound contradictory, it has to be noted that due to the nature of the networks, this optimization is done on RGB values in the original attacks and might not be optimal when YUV domain is considered. The proposed method reduces the noise in U and V channels which is compensated by increasing the noise in Y channel. This strategy reduces the amount of perceptible color noise as well as reducing the total noise as indicated by L_2 distances calculated using RGB channels.

CHAPTER 5

CONCLUSION

This thesis proposed an attack and network type agnostic perceptual enhancement method by converting the adversarial noise to YUV color space and reducing the chrominance noise and applying Gaussian smoothing to the adversarial noise. The adversarial images are not only perceptually better but also have lower L_2 distances to the original images. Conventional networks are trained using images in RGB color space and inherently, the optimization is done in this color space. In the future, these networks could be trained using images in YUV color space. Then using these networks, attacks could be done intrinsically in YUV space.

For future work, Cycle-Spinning GAN, a recently proposed method [34], can be used the craft adversarial images or defensing against adversarial attacks.

The proposed method assumes that the object is located near the center of the image and Gaussian kernel is positioned at the center of the image. However the object could be off-center or could be located in a different position which might invalidate this assumption. In the future, class activation maps [32] or gradient based localization of object [33], which could be obtained directly through the attack network, can be used to estimate the center position of the object. This would allow positioning the Gaussian kernel to overlap better with the object position.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances In Neural Information Processing Systems*, NIPS'12, (USA), pp. 1–9, Curran Associates Inc., 2012.

[2] G. F. Elsayed, S. Shankar, B. Cheung, N. Papernot, A. Kurakin, I. Goodfellow, and J. Sohl-Dickstein, "Adversarial examples that fool both computer vision and time-limited humans," *Advances in Neural Information Processing Systems*, pp. 3914–3924, 2018.

[3] C. Shi, X. Xu, S. Ji, K. Bu, J. Chen, R. Beyah, and T. Wang, "Adversarial CAPTCHAs," *arXiv preprint arXiv:1901.01107*, jan 2019.

[4] I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet, "Multi-digit number recognition from street view imagery using deep convolutional neural networks," *International Conference on Learning Representations*, 2014.

[5] F. Stark, C. Hazırbaş, R. Triebel, and D. Cremers, "Captcha recognition with active deep learning," *GCPR Workshop on New Challenges in Neural Computation*, vol. 10, 2015.

[6] S. Sivakorn, I. Polakis, and A. D. Keromytis, "I am robot: (deep) learning to break semantic image captchas," in *2016 IEEE European Symposium on Security and Privacy (EuroS P)*, pp. 388–403, March 2016.

[7] I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet, "Multi-digit number recognition from street view imagery using deep convolutional neural networks," *International Conference on Learning Representations*, 2014.

[8] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[9] N. Papernot, P. D. McDaniel, I. J. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against deep learning systems using adversarial examples," *arXiv preprint arXiv:1602.02697*, vol. abs/1602.02697, 2016.

[10] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples," *arXiv preprint arXiv:1605.07277*, 05 2016.

[11] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, pp. 1–20, 01 2019.

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

[13] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *arXiv preprint arXiv:1607.02533*, vol. abs/1607.02533, 2016.

[14] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," *A Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9185–9193, 2016.

[15] S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," *arXiv preprint arXiv:1511.04599*, vol. abs/1511.04599, 2015.

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

[17] R. C. Gonzalez and R. E. R. E. Woods, *Digital image processing*. Prentice Hall, 2008.

[18] V. Viqueira Pérez, D. De Fez Saiz, and F. Martinez Verdú, "Colour vision: theories and principles," *Colour Measurement*, pp. 3–e2, jan 2010.

[19] J. M. Kasson and W. Plouffe, "An analysis of selected computer interchange color spaces," *ACM Transactions on Graphics*, vol. 11, pp. 373–405, oct 1992.

[20] A. E. Aydemir, A. Temizel, and T. T. Temizel, "The effects of JPEG and JPEG2000 compression on attacks using adversarial examples," *arXiv preprint arXiv:1803.10418*, 2018.

[21] Kaggle, "Nips 2017: Adversarial learning development set," Jul 2017.

[22] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *ICLR Workshop*, 2017.

[23] 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, R. Long, and P. McDaniel, "Technical report on the cleverhans v2. 1.0 adversarial examples library," *arXiv preprint arXiv:1610.00768*, 2016.

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

[25] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pp. 4278–4284, 2017.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[27] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, pp. 600–612, April 2004.

[28] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Transactions on Image Processing*, vol. 21, pp. 4695–4708, Dec 2012.

[29] M. A. Saad, A. C. Bovik, and C. Charrier, "Dct statistics model-based blind image quality assessment," in *2011 18th IEEE International Conference on Image Processing*, pp. 3093–3096, Sep. 2011.

[30] A. Mittal, R. Soundararajan, and A. C. Bovik, "Making a "completely blind" image quality analyzer," *IEEE Signal Processing Letters*, vol. 20, pp. 209–212, March 2013.

[31] A. Mittal, A. K. Moorthy, and A. C. Bovik, "Blind/referenceless image spatial quality evaluator," in *2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, pp. 723–727, Nov 2011.

[32] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2921–2929, 2016.

[33] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, "Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization," *CoRR*, vol. abs/1610.02391, 2016.

[34] U. Uzun and A. Temizel, "Cycle-spinning gan for raindrop removal from images," *IEEE Int. Conf. Advanced Video and Signal Based Surveillance (AVSS)*, 2019.

[35] S. Li, I. B. Araujo, W. Ren, Z. Wang, E. K. Tokuda, R. H. Junior, R. Cesar-Junior, J. Zhang, X. Guo, and X. Cao, "Single image deraining: A comprehensive benchmark analysis," *CoRR*, vol. abs/1903.08558, 2019.

APPENDIX A

ADDITIONAL EXPERIMENTS

Human visual system is more sensitive to luminance channel than chrominance channel as explained in Sec. 2.4. As an additional experiment, thinking the HVS, small changes were made in the Alg.1, while the noise in Y channel is scaled by a factor of α, the noise in U and V channels isn't scaled. The second version of Alg.1 is shown below:

Algorithm 3 Iteratively Finding the Minimum Adversarial Noise (Reverse of Alg.1)

Convert the original image $I^{R,G,B}$ into YUV: $I^{Y,U,V}$
Initialize the best distance L_2' to a high number
while Attack is successful **do**
  Run the attack to generate adversarial noise image $N^{R,G,B}$
  Convert $N^{R,G,B}$ into YUV: $N^{Y,U,V}$
  Scale the noise in Y channel by a factor of α, apply Gaussian smoothing G to all noise channels and construct the adversarial image:
  $A^Y = I^Y + G(\alpha \times N^Y)$
  $A^U = I^U + G(N^U)$
  $A^V = I^V + G(N^V)$
  Convert $A^{Y,U,V}$ into RGB: $A^{R,G,B}$
  Calculate the new distance L_2 using $A^{R,G,B}$ and $I^{R,G,B}$
  if $L_2 < L_2'$ and attack is successful **then**
    Store the best attack:
    $A'^{R,G,B} = A^{R,G,B}$
    Store the minimum L_2 value as the new minimum
    $L_2' = L_2$
    Decrease the attack strength (ϵ for FGSM, MIM and BIM, maximum iteration for C&W L_2 and DeepFool)
  else return $A'^{R,G,B}$
  end if
end while

Alg.3 has been trained to find the best α value on the training set by optimizing L_2 distance. However Alg.3 should produce better results thinking the human visual system intrinsically, scaling the noise just in Y channel isn't enough to produce better results than Alg. 1. The proposed method using Alg.1 has produced lower L_2 scores in 84 out of 90 case.