Integrating Single-Shot Fast Gradient Sign Method (FGSM) with Classical Image Processing Techniques for Generating Adversarial Attacks on Deep Learning Classifiers

Muhammad Hassan[a], Shahzad Younis[a], Ahmed Rasheed[a], and Muhammad Bilal[b]

[a]School of Electrical Engineering and Computer Science, National University of Sciences and Technology (NUST), Islamabad, Pakistan
[b]Faculty of Business and Law, University of the West of England (UWE), Bristol, United Kingdom

ABSTRACT

Deep learning architectures have emerged as powerful function approximators in a broad spectrum of complex representation learning tasks, such as, computer vision, natural language processing and collaborative filtering. These architectures bear a high potential to learn the intrinsic structure of data and extract valuable insights. Despite the surge in the development of state-of-the-art intelligent systems using the deep neural networks (DNNs), these systems have found to be vulnerable to adversarial examples produced by adding a small-magnitude of perturbations. Such adversarial examples are adept at misleading the DNN classifiers. In the past, different attack strategies have been proposed to produce adversarial examples in the digital, physical, and transform domain, but the likelihood to generate perceptually realistic adversarial examples require more research efforts. In this paper, we present a novel approach to produce adversarial examples by combining the single-shot fast gradient sign method (FGSM) and spatial, as well as, transform domain image processing techniques. The resulted perturbations neutralize the impact of low-intensity based regions, thus, instilling the noise only in the selective high-intensity regions of the input image. While combining the customized perturbation with one-step FGSM perturbation in an un-targeted black-box attack scenario, the proposed approach successfully fools state-of-the-art DNN classifiers with 99% adversarial examples being misclassified on the ImageNet validation dataset.

Keywords: FGSM, Image Processing, Steganography, Perturbations, Adversarial Examples, Black-Box Attacks.

1. INTRODUCTION

The past decade has witnessed a widespread use of deep neural networks (DNNs) in multi-disciplinary real-life applications ranging from medicine [1], to self-driving cars [2], and to natural language processing domain [3]. These algorithms have found their way to influence life-critical decision making activities by empowering enterprise applications. Despite great successes in numerous applications, recent studies have revealed that intelligent algorithms are vulnerable to mislead predictions by slightly adjusting the input data. For instance, a carefully chosen small perturbation injected in the system's input can cause an unintended behaviour at the output, thus, impeding its functionality. A similar disruption can happen in DNNs, such that, it can push the model to produce adversary-selected results. The situation can lead to the catastrophic results while putting the human lives and the security-critical applications at risk.

In an image classification problem, Szegedy et al. [4] first generated trivial perturbations for input images and fooled the AlexNet classifier with a high probability score. These small perturbations cause no apparent changes to the input images and hard for misleading the human vision apparatus (i.e. eyes), yet, still fool sophisticated algorithms. The manipulated inputs (by adding perturbations) that can mislead the computer vision algorithms are known as adversarial examples. As undeniably exploiting the power of DNNs in various applications, these networks have also created a window of opportunity for the attackers to generate appropriate adversarial examples. For instance, the adversary may require a comprehensive prior knowledge of various potential adversarial techniques to generate imperceptible adversarial examples, or, it can also craft adversarial examples using the techniques that are unknown to the classifier, such as, adaptive noise reduction [5].

Further author information: (Send correspondence to Muhammad Hassan)
E-mail: m.hassan057@gmail.com

The field of adversarial Machine Learning (ML) also focuses on enabling DNNs to safeguard against such attacks. Several studies have been carried out in this direction ranging from detecting adversarial examples at source [6–9] to training resilient DNN models with input data comprising of sufficient adversarial examples, or training auxiliary models for detecting adversarial attacks [6,7] to evaluating DNNs using statistical tests on adversarial and benign inputs [8,9].

The existing research on adversarial ML largely undermines an intrinsic quality of DNNs that these models carry varying information in different parts of the network. Most adversarial example generation approaches introduce perturbation to the entire input image regardless of the importance of individual regions to model predictions. In this paper, we aim to address such limitations and present a hybrid adversarial attack approach to generate unique perturbations using a combination of single-shot FGSM with spatial (entropy or variance) or transform domain (DCT, DWT or FFT) image processing techniques. Fig. 1 illustrates the over-all workflow diagram of our proposed methodology. The main underlying intuition is that adversarial perturbations shall be confined to limited (information-rich) regions of the input image for the adversarial change to remain indistinguishable. Therefore, the first objective is to identify high-intensity regions using classical image processing techniques. We call such regions as "high-profile" regions. These regions tend to carry more importance toward generating adversarial perturbations. The second objective is to exploit these regions to generate highly realistic and imperceptible adversarial examples. Lastly, to evaluate the robustness of our approach to successfully fool state-of-the-art DNN models (i.e., Inception V3, AlexNet, ResNet, VGG16) using the generated adversarial examples. A summary of the approach and key contributions of this paper are as follows:

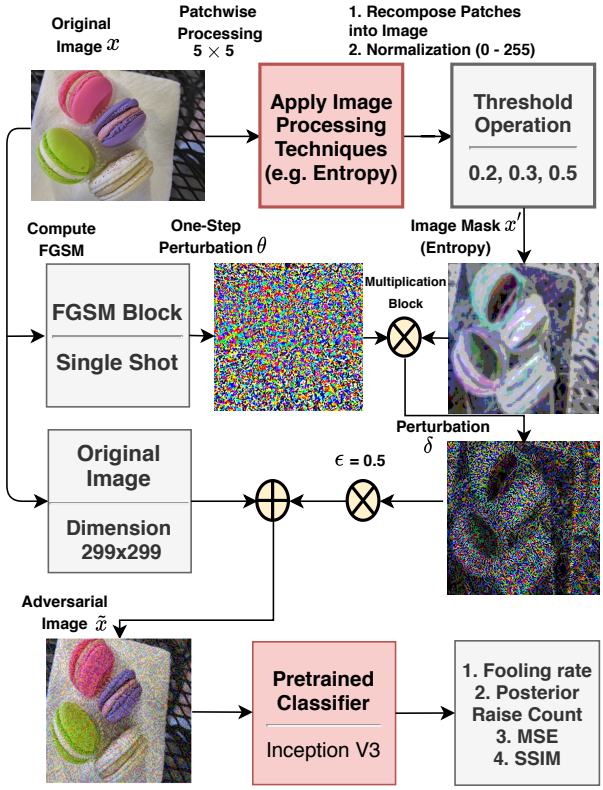

Figure 1. Forward pass of entropy-based adversarial example using fix epsilon in each channel of host image.

- By applying spatial (or transform domain) image processing techniques on the host image, mask image entailing high-intensity region is obtained. These masks tend to induce perturbations only in the selected (high varying) regions when combine with single-shot FGSM based perturbations.
- While incorporating the single-shot FGSM technique, the proposed approach incorporates a fixed epsilon ϵ value, as well as, a different ϵ value for each host image channel (RGB). The mask image (e.g., entropy

mask), when combines with the variable ϵ based FGSM perturbation produces more imperceptible adversarial examples as compared to the fixed ϵ based adversarial examples. At the high-category threshold of 0.7, the SSIM value (averaged) of 0.88 is obtained between the host images and the adversarial examples generated using the FGSM perturbation with an entropy mask.

- The pre-trained classifiers are vulnerable to adversarial attacks as the proposed techniques (i.e., instilling the FGSM perturbation with spatial and transform-domain perturbation masks) are adept at fooling state-of-the-art DNN classifiers. In case of AlexNet classifier, 99% adversarial examples are misclassified, whereas, the InceptionV3 classifier successfully fools on average 70% of the generated adversarial examples.

The rest of the paper is organized as follows. Section 2 outlines the background information in context of single-shot FGSM and spatial and transform-domain image processing techniques. Section 3 explains the methodology for (1) generating image masks (2) attack algorithm using fix epsilon ϵ, and (3) attack algorithm using variable epsilon ϵ. Section 4 and 5 reports the experimental setup and results, respectively. Finally, Section 6 concludes the paper with a discussion.

2. BACKGROUND

While machine learning systems are working at par, adversarial ML [10] has jeopardized the security and privacy of smart intelligent systems. These systems have become vulnerable to the latest adversarial attack methodologies, such as training data-unaware imperceptible security attacks [11] and decision-based attacks [12,13] that extract tangible information to weaken the security of these systems. For instance, the attacker can target a machine learning system during the learning phase to temper with the training data, or it can manipulate the data on which the model is making predictions during the inference time. In such a scenario, understanding the behaviour of systems and the learning algorithms is an important line of work. Among the perturbations generated for inference attacks, a class of adversarial inputs, known as adversarial examples was introduced by [4]. The indistinguishable perturbations when injected in state-of-the-art DNN architectures have revealed their serious limitations which led the ML research to quest for a series of follow-up work. The research shows that the perturbations could be produced with minimal knowledge [14], and it can also be engineered using methods from the digital domain, physical domain [15,16] and most recently the transform-domain [17].

While the first demonstration of adversarial examples [18] has led to slanted attack methodologies, this section explains the technical details on generating adversarial examples using the first-order FGSM approach and few basic image processing techniques that are implemented using spatial and transform domain methodologies.

2.1 Overview of Fast Gradient Sign Method (FGSM) for Adversarial Attacks

2.1.1 Notations:

Let us say we have an input sample pair (x, y) which belongs to a set of training samples $\mathbb{X}$ (i.e., $x \in \mathbb{X}$). Let $F(\cdot)$ denotes a DNN based classification model (pre-trained) which takes the vector representation of an input image x (host image) and outputs the probability distribution $Z(\cdot)$ (i.e., logits) over all possible classification labels. Assuming, a softmax activation function is applied on the logits, the predicted label of the host image x is the one with the highest estimated probability. Formally, it can be expressed as:

$$C(x) = \underset{i}{argmax}(F(x)_i), \tag{1}$$

where $F(x) = softmax(Z(x))$.

2.1.2 Attack Algorithm:

In the past, various first-order algorithms have been proposed to generate adversarial examples, such as, single-step Finite Gradient Sign Method (FGSM) [18], the Randomized Fast Gradient Sign Method (RAND+FGSM) [19], the iterative version of FGSM, i.e., Projected Gradient Descent (PGD) [15]. For a host image x, these methods add perturbation in the direction of gradient of loss function to generate independent adversarial examples.

Given an input image x, the goal is to generate adversarial example $\widetilde{x}$, which should be perceptually indistinguishable from the input x. Theoretically, this is obtained by adding an adversarial perturbation θ to the original input x. As a result of such perturbations, the true class label of the host image x is changed to a different class label, i.e., $C(x_i) = C(x_j)$,

where $i \neq j$. Note that, we want the adversarial example to be indistinguishable from the original one. Thus, adversarial examples should be constrained by the magnitude of adversarial perturbation as $\|x_i - \widetilde{x}_i\|_\infty \leq \epsilon$, i.e., the L_∞ norm should be less than the epsilon ϵ value. Here, L_∞ denotes the maximum change for all pixels of an adversarial example.

FGSM is a fast and computationally efficient method to generate adversarial examples. It performs single step gradient update on the original samples along the direction of the gradient of the loss function $\mathcal{L}(x, y; \theta)$. The loss function is usually defined as the cross-entropy between the output of a classifier and the true label y_i. Formally, the adversarial examples using FGSM approach are expressed as:

$$\widetilde{x}_i = \textbf{clip}_{[0,1]}\{x_i + \epsilon \cdot sign\nabla_x \mathcal{L}(x_i, y_i; \theta)\}, \tag{2}$$

where ϵ controls the maximum L_∞ perturbation of the adversarial examples (i.e., magnitude), and the $\text{clip}_{[a,b]}(x)$ forces the sample x_i to reside in the range of $[a, b]$.

2.2 Spatial-Domain Image Processing Techniques

Variance: The first image statistic that we used to evaluate the image characteristics is the local variance of image intensity (i.e., the second central moment of the pixel intensities within a local neighborhood, thus, the "local variance" hereinafter). Computing the local variance helps in measuring the visual saliency and smoothness in an image. Local variance has been applied to many areas of image processing and analysis. For instance, in the domain of image quality assessment, Lai-Man Po [20] shows that image patches with complex structures have much higher changes of achieving better image quality score. Similarly, in image filtering, local variance has been applied as a measure to control the degree of filtering on local regions. Formally, let I denote an M×N block of a grayscale image. The local variance, i.e., the second central moment of pixel intensities, is defined as:

$$var(x) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (I_{m,n} - u)^2, \tag{3}$$

where μ denotes the mean pixel intensity at the I^{th} block.

Entropy: Entropy is a measure of image information content which is interpreted as the average uncertainty of information source. In an image, discrete entropy is defined as corresponding states of intensity level that individual pixels can adapt. It is a powerful metric and has widely been used in many image processing tasks. Without loss of generality, for an M×N grayscale image with 256 pixel values (0∼255), the function entropy takes a 1-dimensional array and calculates the entropy of the pixels in that array, expressed as under:

$$H(x) = -\sum_{i=0}^{255} p_i log_2(p_i), \tag{4}$$

where $p_i = f_i/M \times N$ is the frequency of i^{th} pixel level.

2.3 Transform-Domain Image Processing Techniques

Discrete Cosine Transform (DCT) Given that, the recent work has examined adversarial examples in the digital [21], as well as, in the physical domain [22], adversarial perturbations in transform domain can also exploit the vulnerability of DNNs [17]. Thus, carrying the similar notion, transform domain techniques are of significant importance in understanding the image characteristics. Given I as an M×N block of a grayscale image matrix, the two-dimensional DCT is defined as:

$$\begin{aligned} F(u, v) = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I_{m,n} \\ \cos(\frac{\pi(2m+1)p}{2M}) \cos(\frac{\pi(2n+1)q}{2N}), \end{aligned} \tag{5}$$

where the values $F(u, v)$ are called the DCT coefficients of image pixels at the coordinate (m, n), and α_p, α_q are the basis functions.

Discrete Wavelet Transform (DWT) In a digital image, edges contain the most important high-frequency information. To extract such information, discrete wavelet transform (DWT) is generally considered that split the input image into

four non-overlapping sub-bands of frequency, i.e., LL (low-low), LH (low-high),HL (high-low), and HH (high-high). The LL (approximation-component) sub-band represents the coarse-scale DWT coefficients, whereas, the sub-bands LH (horizontal component), HL (vertical component) and HH (diagonal component) reflect the fine-scale of DWT coefficients. In literature, researchers have used DWTs for performing steganography, as well as, to generate adversarial examples using the approximate and fine-scale DWT components [17,23]. Without loss of generality, let I denote the M×N block of a grayscale image. The two-dimensional DWT of the I^{th} block is defined as:

$$W_{\psi}^{i}(j,u,v) = \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I_{m,n} \phi_{j_0,u,v}(m,n), \tag{6}$$

where $i = \{H, V, D\}$. $W_{\psi}(j,u,v)$ coefficients represent horizontal, vertical, and diagonal components of the block $I_{m,n}$.

Fast Fourier Transform (FFT) The use of fourier analysis is beneficial for variety of image processing techniques, such as, image manipulation, filtering, correction and compression. The technique has been adopted to perform image steganography [24] and generating adversarial examples [25]. For an M×N block of a digital image, the two-dimensional FFT is defined as follows:

$$F(u,v) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I_{m,n} e^{-j2\pi} (\frac{mu}{M} + \frac{nu}{N}), \tag{7}$$

where $I_{m,n}$ is the corresponding image pixel value of block size M×N.

3. METHODOLOGY

3.1 Overview

The proposed approach aims to generate a perturbation by employing spatial and transform domain image processing techniques, and embed it with a single shot FGSM perturbation to generate the adversarial examples. The idea behind it is to modify the traditional FGSM based perturbation technique and incorporate the perturbations of only high intensity or sharp transition based regions of a host image. This can be accomplished using the techniques of image processing, such as, variance, entropy as well as, the transform domain methods, i.e., DWT, DCT, and FFT. For example, computing variance in a selective block (e.g., block size of 5×5) of a host image shall separate out the high and low intensity regions. In our work, we leverage such intensity variations and introduce a threshold constraint that clips out the image pixel values that fall below the threshold constraint. Hence, for a host image, image processing techniques tend to reduce the overall adversarial effect by superimposing only the high varying pixel values to the traditional FGSM based perturbation technique.

3.2 Threat Model

In this paper, we propose to incorporate a common threat scenario where the adversaries can attack the victim model only at the testing or the deployment stage. After the victim model has been trained, the attacker can tamper the input data to generate the adversarial samples and estimate the strength of the model against the embedded perturbation. Furthermore, the proposed scenario also assumes that the adversaries may have the prior knowledge of the trained models (i.e., the architectures and parameters) but are not allowed to modify them. Later on, against these adversarial examples, the model's integrity (i.e., the *adversarial goal*) is calculated by employing the performance metrics, such as, the fooling rate value, posterior raise count, mean square error (MSE), and structural similarity index measure (SSIM).

3.3 Generating Image Masks

In an image dataset, the components of high intensity region play an important role in generating adversarial perturbations. In this sub-section, we formulate a methodology to generate the mask image x_i' of a clean host image x_i using the proposed image processing techniques. The idea is to decompose a host image x_i of dimensions 229×229 ×3 into non-overlapping patches of size 5×5. For each patch, we compute variance (or entropy, DCT, DWT and FFT). Later on, these patches are sequentially recombined to obtain an intermediate image of dimensions 229×229×3. The intermediate image is linearly normalized between the range of 0 to 255, and passes through the threshold block T to obtain the final mask image x_i' of the host image x_i (dimensions 229×229×3). For each technique, we set three different threshold values, categorized as,

Table 1. Selected thresholds in each category of spatial and transform-domain image processing technique.

Methods		Category		
		Low	**Medium**	**High**
Spatial Domain	**Variance**	0.1	0.15	0.2
	Entropy	0.2	0.5	0.7
Transform Domain	**DCT**	0.3	0.5	0.7
	DWT	0.3	0.5	0.7
	FFT	0.1	0.2	0.3

low, *medium* and *high*. Table 1 shows the selected threshold values. Furthermore, the steps to obtain the mask image x_i' in each technique are enlisted below:

- **Variance**: For each (RGB) channel of host image x_i, compute the variance (Eq. 3) value $var(x_i)$ of every 5×5 non over-lapping patch. The response from each channel is combined together to obtain the mask image x_{var}.
- **Entropy**: For each (RGB) channel of host image x_i, compute the discrete entropy score (Eq. 4) $H(x_i)$ of non-overlapping 5×5 patch. The score of each channel is combined together to obtain the final mask image x_{ent}.
- **DCT**: For each (RGB) channel of host image x_i, apply the two-dimensional discrete cosine transform (DCT) (Eq. 5) on non over-lapping 5×5 blocks (patches). The blocks are combined together from each channel to obtain the mask image x_{DCT}.
- **DWT**: For each (RGB) channel of host image x_i, apply the level-1 two-dimensional discrete wavelet transform (DWT) (Eq. 6) on non over-lapping 5×5 blocks (patches). For each patch, extract only the diagonal information (HH) sub-band and recombine the patches from each channel to obtain the respective mask image x_{DWT}.
- **FFT**: For each (RGB) channel of host image x_i, compute the fast fourier transform (FFT) (Eq. 7) on non over-lapping 5×5 patches. For each patch, extract only the phase component, and recombine the patches from each channel to obtain the corresponding mask image x_{FFT}.

3.4 Attack Algorithm using FGSM (Fix ϵ)

For exploratory purposes, we select FGSM to construct adversarial examples, as the method is computationally cheap and linearly approximates the DNN classifiers. The ability to solve the maximization problem in a closed form still makes it a reliable technique to fool pre-trained networks in the context of computer vision classification [26]. In this sub-section, we highlight the significance of the proposed mask images by devising the methodology to generate perceptually better adversarial examples. The presented approach generates FGSM perturbation in a one-shot attack manner. Using chain rule, each pixel of a host image x contributes to the loss value $\mathcal{L}(x, y; \theta)$, and hence, the method finds the required gradients with respect to the host image. Once the gradients are acquired from FGSM, we generate the newly adversarial perturbation by multiplying the mask image x' with the FGSM based perturbation (gradients). The generated adversarial perturbation δ, when combines with the original image x tend to produce perceptually realistic adversarial examples. Using Eq. 2, the adversarial example $\widetilde{x}$ can be expressed as:

$$\widetilde{x} = \textbf{clip}_{[0,1]}\{x + \epsilon \cdot \delta\}, \tag{8}$$

where $\delta = (sign\nabla_x \mathcal{L}(x, y; \theta)) \cdot x'$ denotes the customized perturbation achieved by superimposing the mask image x' on the FGSM based perturbation θ, and ϵ is the fixed epsilon value controlling the perturbation's amplitude.

3.5 Attack Algorithm using FGSM (Variable ϵ)

The advantage of instilling perturbations in high intensity regions encourage the DNN classifiers to perform misclassification with a high attack success rate. However, this can be explored further while varying the epsilon value for each individual channel of the host image x. Like the previous presented approach with a fixed ϵ value, the proposed approach follows the similar methodology and generates perturbations by addressing the high intensity regions of FGSM induced

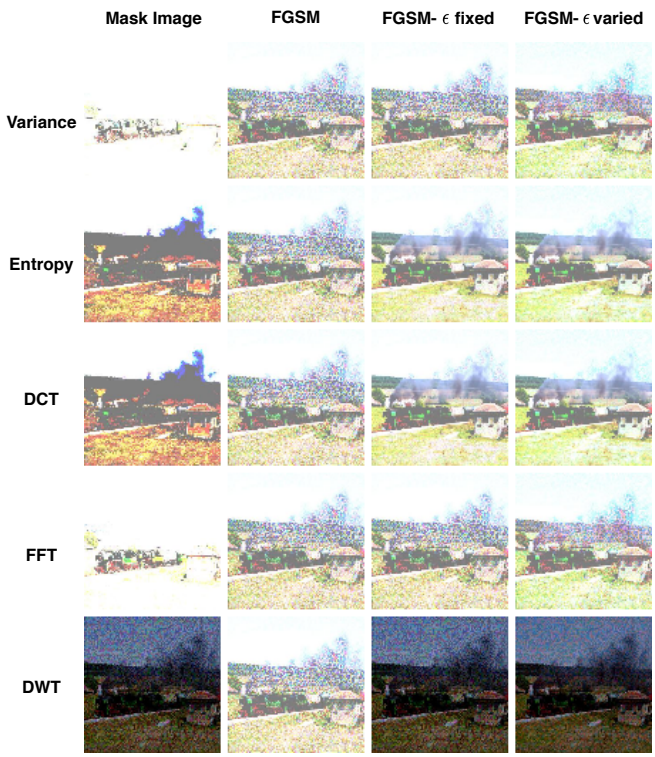

Figure 2. Generated adversarial examples(high-category threshold T) by combining FGSM with the proposed spatial and transform-domain image processing techniques. The adversarial examples are generated using the single-step FGSM, FGSM with ϵ fixed value and FGSM with different ϵ value in each host image channel.

perturbations. Later on, it assigns a different ϵ value for each RGB host image channel which signifies the change in the pixel value of the individual channel (e.g. Red, Green, or Blue). Using Eq. 2, the generated adversarial examples $\widetilde{x}$ can be expressed as:

$$\widetilde{x} = \textbf{clip}_{[0,1]}\{x + ((\epsilon_R \cdot \delta_R) + (\epsilon_G \cdot \delta_G) + (\epsilon_B \cdot \delta_B))\}, \tag{9}$$

where $\delta = (sign\nabla_x \mathcal{L}(x, y; \theta)) \cdot x'$ is the customized perturbation achieved by superimposing the mask image x' on the FGSM based perturbation θ. The epsilon value ϵ controls the amplitude of an individual perturbation channel (red, green, and blue) by assigning a different weight value to the three channels. Fig. 2 shows the adversarial examples that are generated using the proposed methodology.

4. EXPERIMENTAL SETUP

To demonstrate the effectiveness of our proposed methodology, we evaluate the adversarial examples in a black-box attack manner, and demonstrate that adversarial examples generated using fixed and variable epsilon case achieves a near equivalent fooling rate value of the classical FGSM technique. We used PyTorch library to implement our methodology. The experiments have been conducted on AMD Ryzen 2700X with a GeForce GTX 1060 compatible GPU.

4.1 Dataset

In our experiments, we use ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012 validation dataset [27] that has widely been used in the research of object classification and localization algorithms. The dataset comprises of 50,000 colored images that are evenly split (50 objects per category) among 1000 different object categories from different animal species, house-hold objects, vehicle types, and much more.

Table 2. Summary of fooling rate of adversarial examples generated by masking the spatial domain techniques on FGSM perturbation (fixed epsilon ϵ).

Models		**Fooling Rate**			
		Variance		**Entropy**	
	Classical FGSM	**FGSM**	**FGSM**	**FGSM**	**FGSM**
		Fixed ϵ	**Variable ϵ**	**Fixed ϵ**	**Variable ϵ**
InceptionV3	78.12	74.84	72.51	72.78	70.61
AlexNet	99.74	99.64	99.47	99.46	99.24
ResNet18	98.74	98.12	97.63	97.20	96.51
VGG16	97.13	96.09	94.65	94.72	93.11

4.2 Models

To perform un-targeted black-box attacks on the three proposed methodologies, we incorporate four famous pretrained models, i.e. InceptionV3 [28], AlexNet [29], ResNet18 [30], and VGG-16 [31]. All these models have achieved classification accuracy while competing the ILSVRC challenge.

4.3 Adversarial Goal

The object of the adversary is inferred from the incorrectness of the model. Based on the impact on the classifier output integrity, Using our proposed methodology, the adversarial goals are defined as under:
Un-targeted Misclassification: For a particular DNN classifier F (e.g., InceptionV3), our first adversarial goal is to change the classification output of a host image x to any class different from the original class. To evaluate the effectiveness of the proposed goal on the pre-trained classifiers, we compute and compare the class labels of host images x_{cl} (validation samples) with the class labels of adversarial examples $\widetilde{x}_{cl}$. After that, fooling rate F_R is computed by counting the number of adversarial images whose class labels do not match with the class labels of host images.
Confidence Reduction Assuming, the pre-trained classifier correctly classifies the class label of a host image x and adversarial image $\widetilde{x}$ (i.e., class labels do not change), our second adversarial goal is to evaluate whether the DNN classifier classifies the adversarial image $\widetilde{x}$ to the same class as of host image x, but with a lower posterior probability score (confidence). In this way, we count those number of adversarial images which result in lower confidence score. This helps in evaluating the robustness of our proposed methodologies against the classical FGSM based perturbation technique.

5. EVALUATION RESULTS

For each image in the ImageNet validation dataset, we compute three adversarial images using the classical FGSM method, the proposed FGSM with fixed ϵ, and FGSM with variable ϵ methods respectively. These adversarial images are passed through the pre-trained DNN classifiers, and further evaluated according to the defined adversarial goals.

5.1 Evaluating Adversarial Examples using Spatial-Domain Image Processing Technique

Table 2 shows the successful fooling of adversarial examples that are misclassified using the variance and entropy based FGSM perturbation technique. The threshold value for variance mask, as well as, the entropy mask is kept at 0.1 and 0.2 respectively. The fooling rate values are compared with the classical FGSM technique. While the other pre-trained classifiers (i.e., AlexNet, ResNet18, VGG16) are vulnerable to adversarial attacks, it can be observed that, the InceptionV3 model is slightly robust, as 78% adversarial examples are classified into other classes. This is due to the deep 48 convolutional layer architecture of InceptionV3 model which makes it better in differentiating the real host image and its adversarial counterpart. Contrarily, the other pre-trained models have less than the half the number of convolutional layers which make them vulnerable to a slight adversarial perturbation induced in the host image. From table 2, it can be seen that, for every perturbation case, more than 90% of the generated adversarial examples are misclassified when the classical FGSM perturbation technique is combined with variance and entropy mask.

Using the InceptionV3 classifier, while employing the FGSM based perturbation technique, 21.88% adversarial examples attain the similar class labels as of the original host images, but, 20.08% of these examples achieve a lower confidence

Table 3. Summary of fooling rate of adversarial examples generated by masking the the transform domain methods on FGSM perturbation (fixed epsilon ϵ).

Models	**Fooling Rate**			
		DCT	**DWT**	**FFT**
	Classical FGSM	**FGSM Fixed ϵ**		
InceptionV3	78.12	70.44	76.12	74.84
AlexNet	99.74	99.16	99.52	99.64
ResNet18	98.74	95.59	98.76	98.12
VGG16	97.13	92.74	97.83	96.09

score (i.e., posterior probability score) in predicting the same class. In case of generating the FGSM perturbation with a variance mask, the model achieves a confidence score of 15.56% for FGSM with a fixed epsilon value, and 14.28% for FGSM with a variable epsilon value in the host image channels. Whereas, using an entropy mask, the InceptionV3 classifier attains a confidence score of 17.16% and 16.77% for FGSM with fixed and variable epsilon value, respectively. The confidence score depicts that, even though, the classifier has classified the host and the adversarial image to the same class, yet, it predicts the adversarial example's posterior probability with a lesser confidence as compared to the host image. Contrarily, the other three pre-trained models (i.e., AlexNet, ResNet18, VGG16) successfully misclassify the adversarial examples, therefore, the confidence reduction scores of these classifiers are not incorporated.

Further on, we also evaluate the MSE and SSIM metric between each host image and its adversarial example generated using the FGSM technique, as well as, the example obtained while applying the variance and entropy mask on FGSM perturbation. For the variance case, it is observed that, at all thresholds (i.e., low 0.1 , medium 0.15, and high 0.2), MSE values do not incur a major change. Using the fixed ϵ value, the average MSE value between the host images and adversarial examples is 0.018, where as, it is 0.013 in case of assigning a different ϵ value to the host image channels. While using the entropy mask in FGSM perturbation, the average MSE value at the high category threshold (i.e., 0.7) is 0.006, where as, at the low threshold (i.e., 0.2), the average MSE value is similar to the simple FGSM perturbation value, i.e, 0.018.

Similarly, at all thresholds (i.e., low 0.1, medium 0.15, and high 0.2), the average SSIM value for the variance-induced fix ϵ case is similar to the average SSIM value of FGSM technique, i.e., 0.62. On the other hand, at threshold 0.2, and fixing a different ϵ value to each host image channel, the generated adversarial examples look perceptually more realistic to the host images, as average SSIM value turns out to be 0.71. Moving further to the entropy case, at low threshold value (i.e., 0.2), the average SSIM difference between the adversarial examples generated using classical FGSM technique and variable epsilon case is only 9%. The attained values are 0.62 and 0.71, respectively. At the high-category threshold (i.e., 0.7), the adversarial examples generated using a varied ϵ signify the perceptual similarity with its host image counterpart, as the average SSIM value is 0.88.

5.2 Evaluating Adversarial Examples using Transform-Domain Image Processing Techniques

The integrity of the pre-trained classifiers is further evaluated using transform-domain perturbation techniques. Following the transform-domain perturbation masking procedure present in section 3.3, the adversarial examples are initially generated by keeping the ϵ value fixed and later on, it is varied in each host image channel. Table 3 and Table 4 shows the fooling rate of adversarial examples that are misclassified by imposing the DCT, DWT, and FFT based masks over FGSM induced perturbations. The fooling rates are compared to the single-shot FGSM technique. Likewise earlier, the inceptionV3 model is resilient towards the generated adversarial attacks, as 78.12% adversarial examples are classified into other classes, whereas, by employing the DWT based masking technique, 76.12% of the samples reside in other classes.

The similar results are obtained in Table 4, where the ϵ value is varied in each channel of the host image. For InceptionV3 model, the adversarial examples generated through the DWT based masking procedure achieves the fooling rate value of 74.70%, whereas, the other DNN classifiers (i.e., AlexNet, ResNet18, VGG16) still remain vulnerable towards the proposed adversarial attacks. These classifiers successfully fool more than 90% of the adversarial examples.

While comparing the perceptibility (i.e., SSIM) of the host images and the adversarial examples generated using the FGSM and other two proposed techniques, the DCT masked FGSM perturbation attains the highest SSIM (average) value, i.e., 0.88 at threshold 0.7 with a variable ϵ in each host image channel. Whereas, the adversarial examples obtained from the other DCT scenarios, as well as, the perturbation techniques (i.e., FFT and DWT) attain almost a similar SSIM pattern.

Table 4. Summary of fooling rate of adversarial examples generated by masking the transform domain methods on FGSM perturbation (variable epsilon ϵ).

Models	**Fooling Rate**			
		DCT	**DWT**	**FFT**
	Classical FGSM	**FGSM Variable ϵ**		
InceptionV3	78.12	68.33	74.70	72.51
AlexNet	99.74	98.84	99.64	99.47
ResNet18	98.74	94.78	98.21	97.63
VGG16	97.13	90.64	95.65	94.65

An average SSIM of 0.62 is obtained for the low-category thresholds and a value of 0.71 is achieved for the high-category thresholds. In case of applying the DCT mask on FGSM perturbation, the best MSE value (averaged) of 0.006 is obtained when we set different ϵ value to each host image channel.

6. CONCLUSION

Generating perceptually realistic adversarial examples can guise the true nature of actual images. Thus, the concealed information can help the attackers to misguide the DNNs, especially in the decision-making process. In this paper, we present a novel approach to produce perceptually realistic adversarial examples. Our insight is that high-intensity regions or edges of an image can be identified using the spatial (variance, entropy) and transform-domain (DCT, FFT, DWT) image processing techniques. These regions tend to produce a mask image x' which can be combined with the single-shot FGSM perturbation to produce a customized perturbation δ. The epsilon ϵ value (i.e., perturbation's amplitude) played a key role in producing imperceptible adversarial examples. It is observed that, the red channel of the generated perturbation δ carries more significance towards producing a clean adversarial image, as we assign different epsilon weights to the adversarial perturbation (i.e., $\epsilon_R = 0.6$, $\epsilon_G = 0.2$, $\epsilon_B = 0.4$). As a result, the perturbation only appears in the selective high-varying regions or at the edges of the adversarial example $\widetilde{x}$. Thus, it helps in maintaining the visual quality of the adversarial images. Experimental results show that our imperceptible attack methodology is effective against un-targeted black-box attacks. Compared to the 78.12% misclassification achieved using the FGSM, the proposed methodology achieves almost a near equivalent fooling rate value, i.e., 74.84% (ϵ fixed), and 72.51 in (ϵ varied) on InceptionV3 classifier. In our future work, we intend to develop approaches for DNNs to quickly identify and cope with such input images while maintaining the overall accuracy. This will lead towards more resilient and robust machine learning.

References

[1] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *nature*, 542(7639):115–118, 2017.

[2] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[3] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[4] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[5] Bin Liang, Hongcheng Li, Miaoqiang Su, Xirong Li, Wenchang Shi, and Xiaofeng Wang. Detecting adversarial image examples in deep neural networks with adaptive noise reduction. *IEEE Transactions on Dependable and Secure Computing*, 2018.

[6] Reuben Feinman, Ryan R Curtin, Saurabh Shintre, and Andrew B Gardner. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*, 2017.

[7] Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. On detecting adversarial perturbations. *arXiv preprint arXiv:1702.04267*, 2017.

[8] Kathrin Grosse, Praveen Manoharan, Nicolas Papernot, Michael Backes, and Patrick McDaniel. On the (statistical) detection of adversarial examples. *arXiv preprint arXiv:1702.06280*, 2017.

[9] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017.

[10] Marco Barreno, Blaine Nelson, Russell Sears, Anthony D Joseph, and J Doug Tygar. Can machine learning be secure? In *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, pages 16–25, 2006.

[11] Faiq Khalid, Muhammad Abdullah Hanif, Semeen Rehman, Rehan Ahmed, and Muhammad Shafique. Trisec: training data-unaware imperceptible security attacks on deep neural networks. In *2019 IEEE 25th International Symposium on On-Line Testing and Robust System Design (IOLTS)*, pages 188–193. IEEE, 2019.

[12] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017.

[13] Faiq Khalid, Hassan Ali, Muhammad Abdullah Hanif, Semeen Rehman, Rehan Ahmed, and Muhammad Shafique. Red-attack: Resource efficient decision based attack for machine learning. *arXiv preprint arXiv:1901.10258*, 2019.

[14] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, 2019.

[15] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

[16] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1625–1634, 2018.

[17] Zakia Yahya, Muhammad Hassan, Shahzad Younis, and Muhammad Shafique. Probabilistic analysis of targeted attacks using transform-domain adversarial examples. *IEEE Access*, 8:33855–33869, 2020.

[18] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[19] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

[20] Lai-Man Po, Mengyang Liu, Wilson YF Yuen, Yuming Li, Xuyuan Xu, Chang Zhou, Peter HW Wong, Kin Wai Lau, and Hon-Tung Luk. A novel patch variance biased convolutional neural network for no-reference image quality assessment. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(4):1223–1229, 2019.

[21] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[22] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

[23] Po-Yueh Chen, Hung-Ju Lin, et al. A dwt based approach for image steganography. *International Journal of Applied Science and Engineering*, 4(3):275–290, 2006.

[24] Tamer Rabie. Digital image steganography: An fft approach. In *International Conference on Networked Digital Technologies*, pages 217–230. Springer, 2012.

[25] Apoorv Tiwari, Akhilesh Pandey, and Mahendra Kumar. Digital image watermarking using fractional fourier transform with different attacks. *International Journal of Scientific Engineering and Technology*, 3(8):1008–1011, 2014.

[26] Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel. Adversarial attacks on neural network policies. *arXiv preprint arXiv:1702.02284*, 2017.

[27] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet lsvrc 2012 validation set (object detection).

[28] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[29] Alex Krizhevsky. One weird trick for parallelizing convolutional neural networks. *arXiv preprint arXiv:1404.5997*, 2014.

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[31] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.