arXiv:2108.09034v1 [cs.CV] 20 Aug 2021

AdvDrop: Adversarial Attack to DNNs by Dropping Information

Ranjie Duan[1,2*] Yuefeng Chen[2] Dantong Niu[3] Yun Yang[1 †] A. K. Qin[1 †] Yuan He[2]
[1]Swinburne University of Technology, Australia [2]Alibaba Group, China
[3]University of California, Berkeley, USA

Abstract

Human can easily recognize visual objects with lost information: even losing most details with only contour reserved, e.g. cartoon. However, in terms of visual perception of Deep Neural Networks (DNNs), the ability for recognizing abstract objects (visual objects with lost information) is still a challenge. In this work, we investigate this issue from an adversarial viewpoint: will the performance of DNNs decrease even for the images only losing a little information? Towards this end, we propose a novel adversarial attack, named AdvDrop, which crafts adversarial examples by dropping existing information of images. Previously, most adversarial attacks add extra disturbing information on clean images explicitly. Opposite to previous works, our proposed work explores the adversarial robustness of DNN models in a novel perspective by dropping imperceptible details to craft adversarial examples. We demonstrate the effectiveness of AdvDrop by extensive experiments, and show that this new type of adversarial examples is more difficult to be defended by current defense systems.

1. Introduction

Deep Neural Networks (DNNs) have demonstrated their outstanding performance across many applications such as computer vision [24] and natural language processing [47]. Though DNNs have great achievement in these tasks, especially in computer vision, they are known to be vulnerable to adversarial examples. Adversarial examples of DNNs were first discovered by Szegedy et al. [42], which are crafted by adding malicious perturbation on clean images to generate undesirable consequences. Various methods have been proposed to generate adversarial examples [20, 32, 7]. Typically, the generated adversarial perturbation is bounded by a small norm ball, which guarantees the resultant images "look like" benign images.

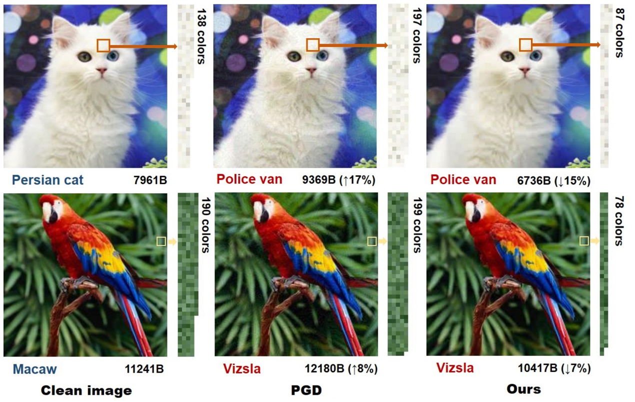


Figure 1: **Adv. images generated by PGD and *AdvDrop*.** Compared to the clean images, the adversarial images generated by *AdvDrop* have fewer details composed of fewer colors, with the decreasing in size (by 15% and 7%).

Interestingly, Ilyas et al. [27] empirically demonstrated that adversarial 扰动 perturbation can be non-robust features for DNNs. That is to say, regarding adversarial perturbation, they are meaningful features for DNNs, but meaningless and imperceptible for humans. So we wonder, whether it is possible to craft 对抗的 adversarial examples in an opposite paradigm? Rather crafting adversarial examples by adding adversarial perturbation (or non-robust features) on clean images, we drop certain features from clean images that are imperceptible to humans but essential for DNNs which further lead to DNNs failing to recognize the resultant images.

Towards this end, we propose a novel adversarial attack named *AdvDrop*, which crafts adversarial images by dropping less perceptible details from clean images. For example, as shown in Figure 1, both adversarial images generated by PGD [32] and *AdvDrop* look indistinguishable from the clean images at first glance. However, when you look closely, PGD generates **extra** details (composed of more colors) at the cost of extra storage (larger image size). In contrast, the proposed *AdvDrop* drops **existing** details such

[*] Work done when Ranjie Duan interns at Alibaba Group, China
[†] Correspondence to: A. K. Qin & Yun Yang
[*] Code is available at https://github.com/RjDuan/AdvDrop

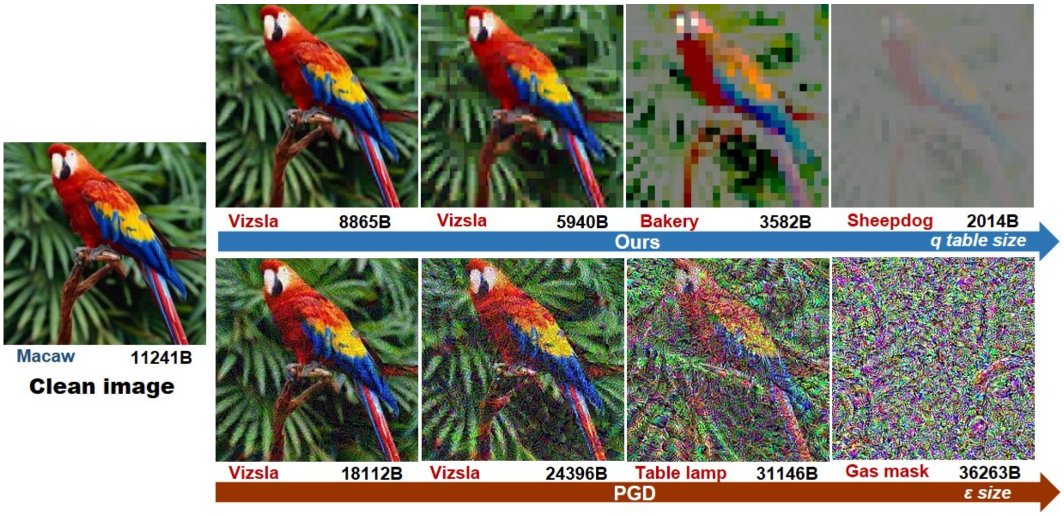


Figure 2: **Interpolation between the clean image and adversarial images generated by *AdvDrop* and PGD.**

as subtle texture-like information from clean images, and the local patch is composed of less colors compared with the other images. As the figure indicates, the lost brittle details from benign images result in DNNs failing to recognize the resultant images correctly.

Dropping information of images can be achieved in either spatial domain (*e.g.* color quantization [25, 34]) or frequency domain (*e.g.* JPEG compression [44]). In our work, we consider developing proposed *AdvDrop* in frequency domain. Principally, we can drop various features of an image to generate adversaries. This preliminary study is focused on the frequency domain because we choose to use "image details" as the feature of interest to be dropped, which can be well quantized in the frequency domain. This choice is motivated by native insensitivity of human eyes to fine image details. In this work, *AdvDrop* first transforms images from spatial domain to frequency domain, then reduces some frequency components of the transformed images quantitatively. Figure 2 shows the process of *AdvDrop*, which performs attack following an opposite mechanism to PGD. The proposed *AdvDrop* starts from removing subtle details (*e.g.* textures) and the resultant image is almost indistinguishable from the clean one. When increasing the amount of dropped information, the resultant adversarial image finally turns to be somewhat "blank". Here,"blank" denotes pure color which presents (almost) no information for recognizing a specific object for DNNs.

We then perform comprehensive evaluation on the proposed *AdvDrop*. It can achieve high attack success rates in both targeted and untargeted settings on ImageNet [12]. We also evaluate the effectiveness of *AdvDrop* in terms of defense methods. Various defense methods have been proposed to defend against adversarial examples [32, 4, 35, 46]. Current defense methods are less effective against adversarial examples generated by *AdvDrop* as they are generated with a rather different paradigm. Moreover, since the adversaries are generated by *AdvDrop* via losing information, they are somewhat robust to denoising-based defenses. Typically, the denoising-based method removes the generated adversarial perturbation and accordingly defends against adversaries (Figure 3). For adversaries generated by *AdvDrop*, however, denoising-based defenses take no effect and the resultant images are still adversarial for DNNs. We hope this finding will motivate devising more effective defense approaches against *AdvDrop*. In addition, to better understand the mechanism of *AdvDrop* and the properties of generated adversarial examples by *AdvDrop*, we provide visualizations of the dropped information by *AdvDrop*, and

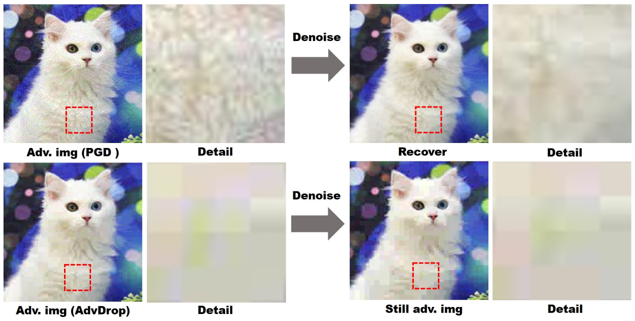


Figure 3: **Adversarial images under denoising-based defense.** Adversarial perturbation generated by PGD could be mitigated by applying denoising strategies, but with almost no effect on adversaries generated by *AdvDrop*.

perform a further analysis together with the attention of the DNNs. In summary, this paper has made the following contributions:

- We propose a novel adversarial attack named *AdvDrop*, which is a totally different paradigm from previous attacks. *AdvDrop* crafts adversarial images by dropping **existing** details of clean images. It opens new doors to generate adversarial attacks for DNNs.

- We conduct comprehensive experiments and demonstrate the effectiveness of *AdvDrop* on targeted and untargeted attack settings. We also empirically show that current defense methods become less effective against adversarial examples generated by *AdvDrop* compared to other attacks.

- Finally, we visualize the dropped information and the attention of the DNNs to interpret the adversaries generated by *AdvDrop*.

This paper is organized as follows. Background and related work are discussed in Section 2. Our proposed approach is described in Section 3, and evaluated in Section 4. Section 5 concludes the paper and points out future work.

2. Background and Related Work

2.1. Adversarial attacks and defenses

Adversarial attack was first proposed by Szegedy et al. [42], aiming to generate perturbation 叠映 superimposed on clean images to fool a target model. Adversarial attacks can be either digital-setting [20, 29, 32] or physical-setting [18, 38, 14], where most attacks are developed in digital-setting. Specifically, given a target model f, adversarial example x' can be crafted by either following the direction of adversarial gradients [20, 29, 32] or optimizing perturbation with a given loss [6, 8]. For most adversarial attacks, the generated adversarial perturbation is bounded by a l_p norm ball. Some works [21, 39] propose generating adversarial examples in the frequency domain for either improving the efficiency in black-box setting or transferability. Roughly speaking, these adversarial attacks can be somewhat formulated as $x' = x + \delta$, where δ represents additive adversarial perturbation. Conversely, the proposed *AdvDrop* crafts adversarial examples with an opposite paradigm $x' = x - \delta$. Note here '+' and '-' are not simple **"add"** or **"subtract"** operations over the values of x, but 表示 denote whether δ is extra information created by attacks or existing information of clean images dropped by *AdvDrop*.

Other works also explore adversarial examples by making modifications or replacement on the secondary attributes (*e.g.* color, lighting [26, 37, 51, 48, 31, 50, 15], texture [45, 14]) to generate adversarial examples. We also note some other works [23, 9] proposed attack based on motion blurring or smoothing, which also lose details of the clean image after the attack. However, their purposes are different from ours. This work aims to demonstrate the effectiveness of the proposed mechanism.

Many adversarial defense techniques have been proposed. Madry et al. [32] proposed adversarial training, which is arguably one of the most effective defense against adversarial attacks. Adversarial training is a data augmentation technique that trains DNNs on adversarial examples rather than natural examples. However, adversarial training is both time and computation consuming, due to the generation of adversarial examples and extra training epochs to fit adversarial examples. There are also preprocessing based methods, which process the inputs with certain transformations to remove the adversarial noise, and then send these inputs to the target model [16, 10, 11, 22, 35]. We consider both types of defenses during the evaluation to evaluate the effectiveness of *AdvDrop*.

2.2. Image compression methods

Image compression methods fall into two categories, lossless compression, *e.g.*, PNG [5], and lossy compression, *e.g.*, JPEG [44, 41]. JPEG compression applies 分离的, 不相关的 discrete cosine transform (DCT) on patches which transforms images from spatial domain to frequency domain. DCT works by separating the image into different parts of different frequencies. Then JPEG applies 量化 quantization matrix (designed based on human vision) on transformed images dropping most of high-frequency components. In detail, higher frequency components of transformed images are rounded to zero, and finally reduce the size of original images. Recently, deep learning based methods have been investigated for image compression problems. Both CNN-based methods [30, 2, 1] and RNN-based methods [43, 28] are investigated. However, deep learning based compression methods are time-consuming and require pre-training. Due to both efficiency and convenience concerns, we follow the design of JPEG to develop our proposed *AdvDrop*.

3. Approach

3.1. Overview

Given a clean image $x \in \mathbb{R}^m$ with class label y, a DNN classifier $f : \mathbb{R}^m \rightarrow \{1, \cdots, k\}$ which maps image pixels to a discrete label set, and a target class $y_{adv} \neq y$ for targeted attack. The goal of adversarial attack is to find an adversarial example x' for clean image x by solving the optimization problem $\mathcal{L}_{adv}(\cdot)$, which is the adversarial loss leading to $f(x') \neq y$ or $f(x') = y_{adv}$. Typically, x' is restricted by l_∞ norm ball: $\|x' - x\|_\infty < \epsilon$. Our goal is to develop a mechanism that drops information from benign images to craft adversarial images. *AdvDrop* is composed

相当于先DCT压缩，丢失一些细节，再IDCT放大. Quantization table是要训练的，能使DCT在压缩时丢失正确的细节，对DNN造成扰动?

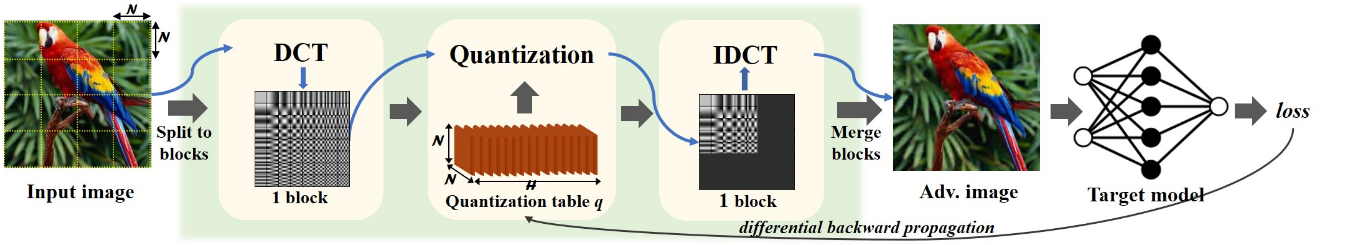


Figure 4: **Pipeline.**

of several parts:

- **Adversarial loss:** The proposed method optimizes over quantization table q by minimizing adversarial loss $\mathcal{L}_{adv}(\cdot)$.

- **Discrete Cosine Transform (DCT):** DCT transforms the input image x from spatial domain to frequency domain. DCT is denoted as $\mathcal{D}(\cdot)$ in the following.

- 反 **Inverse Discrete Cosine Transform (IDCT):** IDCT transforms image's signals from frequency back to spatial domain, denoted as $\mathcal{D}_{\mathcal{I}}(\cdot)$.

- **Quantization:** Quantization serves as the core process to drop information by applying quantization table q, which is optimized during the attack. We denote common quantization as $\mathcal{Q}(\cdot)$. However, in our work, we adopt a differential quantization process denoted as $\mathcal{Q}_{diff}(\cdot)$. Note in the following, either $\mathcal{Q}(\cdot)$ or $\mathcal{Q}_{diff}(\cdot)$ represents a complete quantization-dequantization process.

In summary, our proposed *AdvDrop* first transforms clean images from spatial to frequency domain, then applies quantization to drop some specific frequencies of the transformed image, followed by inverting the frequency signals of images back to spatial domain. During optimization, *AdvDrop* only 调节 tunes the value of quantization table q bounded by ϵ. Formally, we denote our final objective as:

$$\begin{aligned} \min_{q} \ & \mathcal{L}_{adv}(x', y), \text{ where } x' = \mathcal{D}_{\mathcal{I}}(\mathcal{Q}_{diff}(\mathcal{D}(x), q)) \\ & \text{s.t. } \|q - q_{init}\|_{\infty} < \epsilon \end{aligned} \tag{1}$$

where q_{init} is the initial value of quantization table q. We set $q_{init} = \mathbf{1}$. We increase the value of quantization table q gradually to drop the information of given image during the optimization. $\mathcal{Q}_{diff}$ represents a 可微的 differential quantization function, which allows *AdvDrop* to compute the gradients during the backward propagation. The overview of *AdvDrop* is illustrated in Figure 4.

3.2. Adversarial loss

For adversarial loss $\mathcal{L}_{adv}(\cdot)$, we use the following cross-entropy loss:

$$\mathcal{L}_{adv} = \begin{cases} \log(p_y(x')), & \text{for untargeted attack,} \\ -\log(p_{y_{adv}}(x')), & \text{for targeted attack} \end{cases} \tag{2}$$

where $p(\cdot)$ is the probability output (softmax on logits) of the target model f with respect to class y_{adv} or y. By minimizing loss $\mathcal{L}_{adv}$, *AdvDrop* optimizes the quantization table q to selectively drop the information of input image x, in order to mislead the target model f finally.

3.3. Transformation

We introduce both DCT and IDCT in this part. DCT serves as a transformation of images from spatial domain to frequency domain, which expresses a finite sequence of data points in terms of a sum cosine functions oscillating 振荡 at different frequencies. Note that there are also other methods enabling transformation of images from spatial to frequency domain, such as discrete Fourier transform etc. We focus on DCT as it enables the proposed *AdvDrop* to have more flexibility in selecting what and where information to drop.

Before applying DCT, we first split the original images into blocks with size $N \times N$ as shown in Figure 4. Our proposed method splits images into patches with size $N = 8$ [41, 44] for all the experiments by mainly considering computation cost and perceptual quality. As the 相关性 correlation between pixels within the patch decreases with increasing in patch size, the quantization results in larger 失真 distortion. For each block, the value of pixels are adjusted to be symmetric 对称的 with respect to zero. The mathmatical definition of DCT is:

$$\mathcal{D}(x)_{[u,v]} = \frac{1}{\sqrt{2N}} C(u)C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} x[k,m] \\ cos[\frac{(2k+1)i\pi}{2N}] cos[\frac{(2m+1)j\pi}{2N}], \tag{3}$$

In which, Eq. 3 computes the u, v^{th} 分量 entry of $\mathcal{D}(x)$. $x[k,m]$ with the value on coordinate (k,m) of image x. N is the size of the block. A concrete 具体的 example can be seen in Figure 4, where $\mathcal{D}$ transforms input image from spatial domain to a series of blocks in frequency domain.

IDCT is the inverse process of DCT, which serves as recovering 恢复 the signals of input image from frequency back to spatial domain. Due to the page limit, more details can be referred to [3]. Note that either DCT or IDCT is lossless. The information is only lost during the quantization. We then discuss how the quantization is capable of dropping information. 只丢失对DNN有攻击效果的细节

3.4. Quantization

The quantization is done with two operations: rounding and truncation 缩短. The former maps 映射 the original value to its nearest quantization point, while the latter confines 限制 the range of quantized values. A common complete quantization-dequantization process $\mathcal{Q}(\cdot)$ is defined by:

$$\mathcal{Q}(x,\Delta) = \lfloor \frac{x+0.5}{\Delta} \rceil \cdot \Delta,\ \ that\ \ \mathcal{Q}(x,\Delta) \in [\epsilon_{min}, \epsilon_{max}] \qquad (4)$$

where Δ denotes the interval length, which decides the nearest quantization point for value of x, serves as a quantizer. Intuitively, the larger of Δ, the smaller of the length of the set of quantized values after quantization. The quantized values are constrained in a valid range $[\epsilon_{min}, \epsilon_{max}]$.

In our case, we use a trainable quantization table q to quantize the input image x after transformed to frequency domain. Note the purpose of quantization table q is the same as Δ. We increase the amount of dropped details by enlarging the interval of quantization table q. In order to adjust the quantization table q accurately, and further improve the success rate of the proposed attack, we formulate the whole process as an optimization problem by leveraging the gradients of target model f via backward-propagation. However, $\lfloor\cdot\rceil$ is a staircase and thus non-differential function, which cannot be integrated to the optimization via back-propagation directly. To tackle this challenge, inspired by Gong. et al.'s [19] work, we propose a differential asymptotic 渐进的 quantization $\mathcal{Q}_{diff}(\cdot)$ by introducing tangent function into quantization process to approximate the staircase quantization function gradually, such that $\mathcal{Q}_{diff}(\cdot) \approx \mathcal{Q}(\cdot)$. Formally, $\mathcal{Q}_{diff}$ is defined as follows:

$$\mathcal{Q}_{diff}(x,q) = \overset{\text{quantize}(\cdot)}{(\varphi(\frac{x}{q}) + \lfloor\frac{x}{q}\rceil)} \cdot \overset{\text{dequantize}(\cdot)}{q}, \qquad (5)$$

where $\varphi(\cdot)$ approximates the change between two adjacent 相邻的 quantized values. $\varphi(\cdot)$ is continuously differentiable everywhere and defined as follows:

$$\overset{\text{phi\_diff}(\cdot)}{\varphi(\tfrac{x}{q})} = \frac{1}{2}(1+\tanh((\tfrac{x}{q} - \lfloor\tfrac{x}{q}+0.5\rfloor)\cdot\log(\tfrac{2}{\alpha}-1))\cdot\log(\tfrac{1}{1-\alpha})), \qquad (6)$$

where α is an adjustable parameter which controls the steepness 陡峭程度 of slope between two adjacent quantized values. We decrease the value of α linearly to approximate the staircase function gradually during the optimization (Figure 5). The value of α is determined at start and decreased gradually during the optimization, and thus $\mathcal{Q}_{diff}(\cdot)$ behaves almost the same as the desired staircase quantization function at the end of optimization. Due to page limit, more details regarding α can be referred to work [19]. Besides, as the quantization table q should be integers during the optimization, we update the quantization table q with the *sign* of gradients returned via backward propagation. Formally:

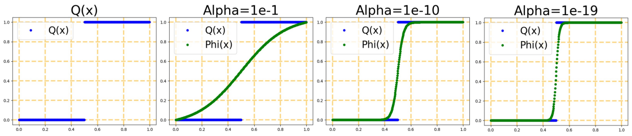


Figure 5: **Illustration of the usage of $\varphi(x)$ and α.**

$$q' = q + sign(\nabla_q \mathcal{L}_{adv}(x', y)),\ \text{s.t.}\ \ \|q - q_{init}\|_\infty < \epsilon \quad (7)$$

where the purpose of ϵ is similar to l_p-norm, aiming to make the resultant adversarial image x' looking indistinguishable from clean image x. In detail, ϵ confines the norm of quantization table q, to further restrict the amount of information to drop. We will give more study about the setting of ϵ in Section 4. The illustration of differential quantization and updation of q at different steps are shown in Figure 6.

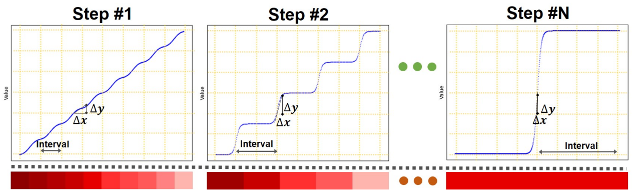


Figure 6: **Differential quantization process and updation of quantization table q.**

During the process of optimization, the interval of q increases gradually as updated by Eq. 7. For illustration, we plot color palettes 色板 at the bottom to show how the colors are dropped when the interval increases. In summary, the proposed *AdvDrop* adopts an asymptotic 渐进的 strategy to drop the information, that the slope ($\frac{\Delta y}{\Delta x}$) becomes steeper gradually, thus $\mathcal{Q}_{diff}(\cdot) \approx \mathcal{Q}(\cdot)$ finally.

4. Experimental Evaluation

We first outline the experimental setup. Then we evaluate our *AdvDrop* regarding its perceptual 感知的 and attack performance. Afterwards, we evaluate the performance of *AdvDrop* under defense methods. We then analyze *AdvDrop* via an 消融 ablation study. We finally analyze the dropped information by $AdvDrop$ together with attention of the model.

4.1. Experimental settings

Dataset and models. We randomly selected 2000 correctly classified images from ImageNet [12] to evaluate proposed attack. We use ResNet50 [24] as the target model for all the experiments. For evaluating the effectiveness of proposed $AdvDrop$ on adversarial training, we used pretrained adversarial model ResNet50 as defense model [17] [†].

Metrics. For all the tests we use attack success rate (succ. rate) (%) as the metric to evaluate the effectiveness of attacks, which is the proportion of successful attacks among the total number of test images defined as $\frac{1}{N}\sum_{n=1}^{N}[f(x) \neq f(x')]$ in untargeted setting, and $\frac{1}{N}\sum_{n=1}^{N}[f(x) = y_{adv}]$ in targeted setting. Regarding evaluation on the visual quality on attacks, we use Learned Perceptual Image Patch Similarity ($lpips$) metric [49] as the perceptual metric.

Baselines. For perception study in Section 4.2, we compared our proposed *AdvDrop* with one of the most commonly used adversarial attack PGD under both l_2 and l_∞ settings with different constraints. For evaluation under various defenses, we consider defense methods including: feature squeezing [46], pixel deflection [35], JPEG compression [40] and adversarial training [32]. Regarding adversarial attacks to compare, we select several state-of-the-art attacks under both l_2 and l_∞ settings, including PGD [32], FGSM [20], C&W [7], DeepFool [33].

4.2. Perception study

We first conduct perception study on the proposed $AdvDrop$. With enlarging the constraint ϵ for quantization table q, the details disappear gradually as shown in Figure 7. We then compare the perceptual quality of adversarial examples generated by *AdvDrop* with other attack methods. Though we have a setting (using ϵ) similar to l_p norm, however, our proposed $AdvDrop$ attack optimizes in the frequency domain, that ϵ is applied as the constraint for q. Therefore we adopt $lpips$ [49] as the perceptual metric, which measures how similar the two images are in a way that 同时 coincides with human judgment. The value of $lpips$ denotes the perceptual loss, the lower, the better. We compare with one of the most commonly used adversarial attack PGD in both l_2 and l_∞ settings.

We summarize the results in Figure 8, where we set *y-axis* with perceptual loss calculated by $lpips$ [49], and set *x-axis* with the change ratio on size of resultant images compared with clean images. For example, for *AdvDrop*-100, the value of *x-axis* represents the size of adversarial images' decreases by 36.32% on average compared with clean images' size. Note that opposite to *AdvDrop*, the size of adversarial images generated by PGD is larger than clean images.

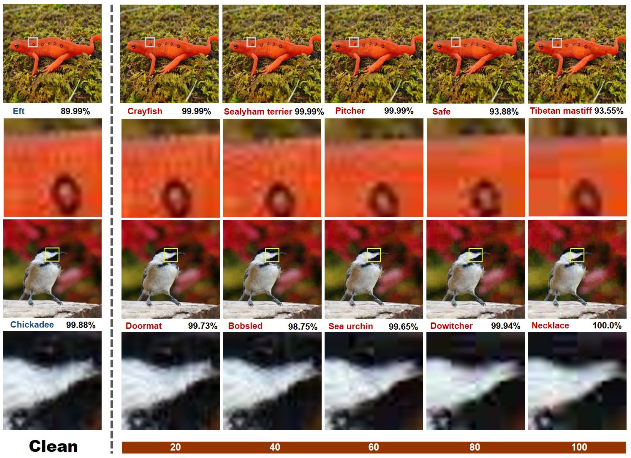


Figure 7: **Adv. images generated by *AdvDrop*.**

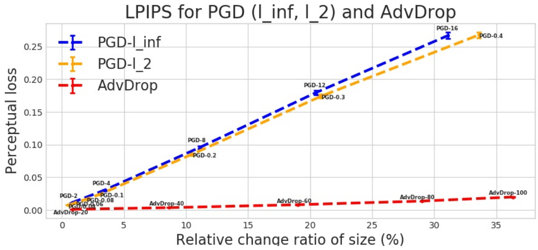


Figure 8: $lpips$ **scores for *AdvDrop* and PGD.**

Thus for PGD, the value on *x-axis* represents how much the ratio in size increases. As Figure 8 indicates, though the relative size ratio changes more compared with PGD in either l_2 or l_∞ settings, the adversarial images generated by $AdvDrop$ are more perceptually aligned 接近,呈直线的 with clean images compared with PGD.

4.3. Evaluation of *AdvDrop*

We now evaluate the performance of *AdvDrop* with both targeted and untargeted settings. We evaluate *AdvDrop* with constraint ϵ for quantization table q with 20, 60, 100 respectively. We summarize the results in Table 1.

Table 1: **Succ. rate (%) of $AdvDrop$ on targeted and untargeted settings with different ϵ.**

ϵ for q	20	60	100
Targeted succ. rate (%)	97.20 ± 0.37	99.45 ± 0.16	99.95 ± 0.05
Untargeted succ. rate (%)	98.55 ± 0.26	99.85 ± 0.08	100.00 ± 0.00

As Table 1 indicates, with relaxing the constraint ϵ, the success rates of *AdvDrop* on both targeted and untargeted

[†] https://github.com/MadryLab/robustness

settings increase. *AdvDrop* can achieve almost 100% success rate when $\epsilon = 100$. We find that *AdvDrop* in targeted setting requires more steps to achieve successful attacks compared with untargeted setting (Figure 9). This may be due to targeted attack always requires more accurate approximation on gradients. During the attack, we set the steps for *AdvDrop* in targeted and untargeted settings with 500 and 50 respectively. We plot the success rates and the loss of a batch on different steps in Figure 9.

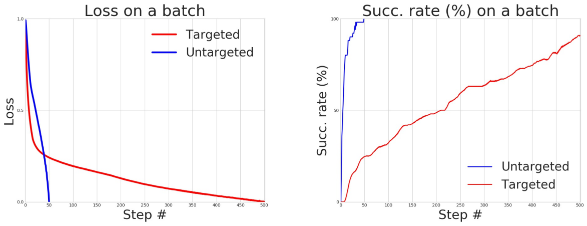


Figure 9: **Loss and succ. rate (%) on a batch.**

As shown in Figure 9, the loss of untargeted attack converges rapidly and achieves 100% success rate when step is around 50. On the other hand, the loss for targeted attack converges until the step is around 500. During the experiments, we find that with the increase of constraint ϵ, fewer steps are required for *AdvDrop* on both targeted and untargeted settings. For example, in a targeted setting, to achieve success rate above 99%, on average 496 steps are required when $\epsilon = 20$, but only 61 steps are required when $\epsilon = 100$.

4.4. Attack effectiveness under defense methods

In this part, we evaluate the effectiveness of proposed *AdvDrop* compared with other adversarial attacks under various defense methods. Here we first generate adversarial examples by adversarial attacks including PGD [32], BIM [13], C&W [7], FGSM [20], and DeepFool [33]. Regarding these attacks, we consider both settings including l_2 and l_∞ to generate adversarial examples. Then we test different defense methods including adversarial training (AT) [4, 32], feature squeezing [46], JPEG compression [40], and pixel deflection (PD) [35] against these samples to evaluate the strength of these attacks under defenses. Among all these defenses, adversarial training is the most effective defense against adversarial attacks. As adversarial training requires too much computation resource, we adopt a black-box setting to evaluate various attacks on adversarial training. We first generate adversarial examples by various attack methods, then feed them to the adversarial trained model. We set $\epsilon = 4$ for attacks on l_∞ setting, set $\epsilon = 0.06$ for attacks on l_2 setting that are common used in previous defense methods [35, 22]. We set quantization table q with 100 for *AdvDrop*. We summarize the results in Table 2.

Table 2: **Succ. rate (%) of attacks under defenses.**

Attacks	No Def.	AT	Feature Squeeze		JPEG-30	PD
			MF-3	Bit-6		
			l_∞			
PGD	100.00	41.60	90.65	70.50	62.50	85.10
BIM	100.00	42.80	90.25	69.20	33.80	81.50
FGSM	91.90	42.60	91.80	66.05	49.60	81.40
			l_2			
PGD	100.0	43.00	90.3	62.8	27.6	64.1
Dfool	99.00	42.95	89.7	29.00	21.20	12.60
CW	84.30	43.00	88.80	26.00	21.70	12.60
AdvDrop	100.00	**44.50**	**95.35**	**82.60**	**80.00**	**95.65**

As Table 2 shows, since adversarial images are crafted by *AdvDrop* with a totally different 范例 paradigm, they are more robust to current defense methods compared to adversarial images generated by other attacks. Among all these defenses, adversarial training is still the most effective defense against *AdvDrop*. We suggest that as the adversarial training makes the model learn more robust feature representations, it is also effective to defend against our proposed *AdvDrop* to some degree. On the other hand, other denoising-based defense methods demonstrate limited effectiveness to defend against our proposed *AdvDrop*. Though denoising-based strategies show effectiveness in 缓和 mitigating the adversarial perturbation generated by previous attacks, regarding *AdvDrop* which has already removed some essential features from clean images, denoising may 使恶化 aggravate the distortion 失真 caused by loss of features. We further perform an evaluation by JPEG compression to validate the robustness of adversarial examples generated by *AdvDrop* under denoising operation.

JPEG compression. Previous studies show that adversarial perturbation can be partly removed via JPEG compression [22, 11]. Regarding JPEG compression, the compression rate is controlled by a quantifiable quality, which affects to what extent the information is reduced. We evaluate the performance of *AdvDrop* under JPEG compression with different quality factors to represent how robust the adversaries generated by *AdvDrop* are. For comparison, we also test the performance of PGD and the accuracy of clean images under JPEG compression (Figure 10). We set both *AdvDrop* and PGD with untargeted setting. We report accuracy rate as the metric in Figure 10, meaning the proportion of recovered adversarial examples by JPEG compression.

As Figure 10 indicates, when quality factor is extreme low (*e.g.* quality factor = 10), JPEG compression results in corruption on clean images, that clean accuracy even drops below 60%. Regarding the adversarial examples generated by PGD, they are mostly recovered when JPEG compression with quality factor being equal to 30. When quality factor further decreases, the recovered rate of PGD decreases due to the corruption caused by JPEG compression. Compared to PGD, adversarial examples generated by *Ad-*

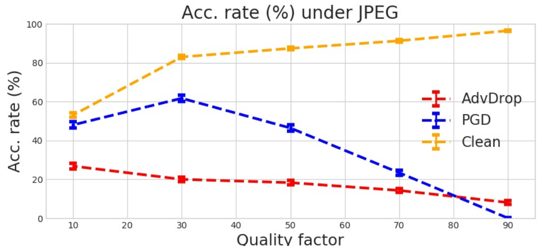


Figure 10: **Performance of *AdvDrop* under JPEG.**

vDrop is much less affected by JPEG compression. Less than 30% adversaries generated by *AdvDrop* are recovered at most. We suggest that the mechanism of *AdvDrop* itself is a kind of lossy operation which makes resultant adversarial images lack some key features for recognition. Thus when further applying lossy operations (such as JPEG compression) on resultant adversarial images, the applied lossy operations may even further destroy the generated adversarial examples and even harder to recover. However, there is still chance that the adversarial examples generated by *AdvDrop* can be recovered by "lossy operation" as Figure 10 shows.

4.5. Ablation study

Here, we conduct experiments to analyze the impact from following aspects for proposed $AdvDrop$ attack: 1) quantization methods, 2) spatial domain, 3) frequency (low frequency, middle frequency or high frequency).

Effect of quantization methods. Here we evaluate how different quantization methods affect the performance of *AdvDrop*. We evaluate a typical rounding method and another differential quantization method proposed by Shin et al. [40]. During evaluation, we only change the quantization method but keep others of *AdvDrop* unchanged. Typical rounding method $\lfloor x + 0.5 \rfloor$ achieves success rate of only 5.00 ± 0.98%. Differential rounding method proposed by Shin et al. [40], $\lfloor x + 0.5 \rfloor$ + $(\lfloor x + 0.5 \rfloor - x)^3$ achieves success rate of 65.20 ± 2.13%, and ours achieves 97.20 ± 0.37% success rate. This demonstrates the effectiveness of differential quantization method adopted by *AdvDrop*.

Effect of frequency domain. Here we perform an ablation study to show the advantage of dropping information in the frequency domain rather than spatial domain. We reduce the color depth to drop images' information. We report the accuracy on the same dataset when reducing the image color by different amounts of bits (Table 3). Compared to dropping information in frequency domain, the accuracy rate is affected until the bit is reduced to 2 (11.10 ± 0.07%). As the images' details could be well quantized in the frequency domain, the resultant images are also more natural for human observers.

Table 3: **Acc. rate (%) by reducing color depth.**

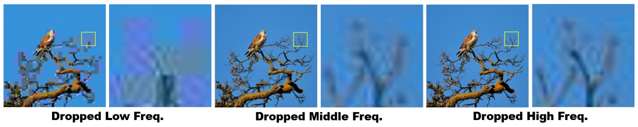
Bit Depth	**2**	**4**	**6**
Acc. rate (%)	11.10 ± 0.07	93.50 ± 0.05	99.90 ± 0.07

Effect of different frequencies. We also perform an ablation study on how dropping different regions of frequency (low, middle, high) affects the model's clean accuracy. We perform this study by dropping different regions of frequency of given images after transformed to frequency domain by DCT. Results are summarized in Table 4. We also visualize the resultant images and their local details at the bottom of Table 4. As the results in Table 4 indicate, compared to middle and high frequencies, the low frequency part serves as dominant feature for the model. When low frequency is dropped from the images, the accuracy rate drops to 30.00 ± 1.02%, but 84.40 ± 8.11% and 86.50 ± 0.76% when middle and high frequencies dropped. As the images in Table 4 show, when low frequency is dropped, the details are almost lost.

丢失高频细节正确率更高

Table 4: **Acc. rate (%) by dropping/reserving different frequencies.**

Dropped Freq.	**None**	**Low**	**Middle**	**High**
Acc. rate (%)	100.00 ± 0.00	30.00 ± 1.02	84.40 ± 8.11	86.50 ± 0.76



4.6. Visualization and analysis

We are also interested in where and what information would be dropped by $AdvDrop$ with a given image? Whether $AdvDrop$ tends to drop the information where models pay attention? Towards this end, we visualize the attention of the model (Grad-CAM [36]) and the amount of dropped information by *AdvDrop* on different regions of given images (Figure 11). Regarding the first case in Figure 11, the model mainly pays attention to the flower part of the "cardoon", *AdvDrop* drops both calyx and flower parts. In the second case, the model pays attention to the head of the "peguin", however, in this case, *AdvDrop* mainly throws away the information on the body part of the "peguin", which has rich texture details regarding the fur of the "peguin". In summary, there is some overlapping between model's attention and where *AdvDrop* drops information from. A small difference is that *AdvDrop* seems

重叠

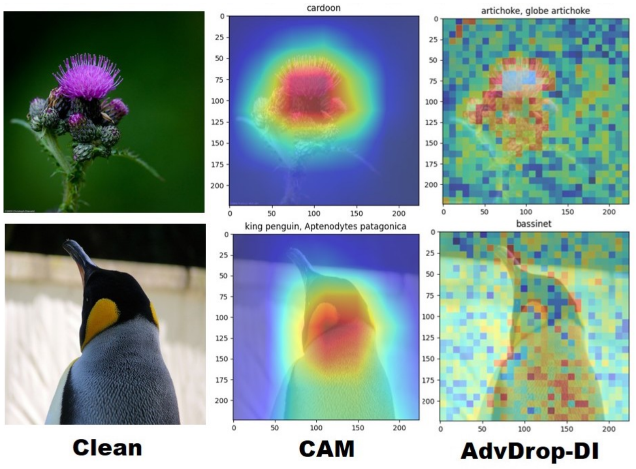

cardoon
artichoke, globe artichoke
king penguin, Aptenodytes patagonica
bassinet
0
25
50
75
100
125
150
175
200
Clean
CAM
AdvDrop-DI

Figure 11: **Analysis on dropped information.**

focusing more on the part which has rich texture details.

We also analyze the components of the dropped information. Here we roughly devide the dropped information as “high frequency” and “low frequency”. We find *AdvDrop* tends to drop high frequency information than low frequency information.

5. Conclusion and Future Work

In this paper, we have investigated the adversarial robustness from a novel perspective, and proposed a novel approach called adversarial drop (*AdvDrop*), which leverages differential quantization and adversarial attack techniques, to craft adversarial examples by dropping existing details of images. *AdvDrop* opens a new way for robustness evaluation of DNNs. The proposed *AdvDrop* currently still utilizes a relative simple method to drop the information by focusing on frequency domain. We plan to explore other techniques to drop the information from images in our future work. Also, we will explore how to apply *AdvDrop* on other tasks such as interpretability of DNNs. Moreover, effective defense strategies against *AdvDrop* will be another crucial and promising direction.

Acknowledgement

This research is partly supported by ARC Discovery Project under Grant DP180100212. This work was supported by Alibaba Group through Alibaba Research Intern Program.

References

[1] Eirikur Agustsson, Fabian Mentzer, Michael Tschannen, Lukas Cavigelli, Radu Timofte, Luca Benini, and Luc Van Gool. Soft-to-hard vector quantization for end-to-end learning compressible representations. *NeurIPS*, 2017. 3

[2] Eirikur Agustsson, Michael Tschannen, Fabian Mentzer, Radu Timofte, and Luc Van Gool. Generative adversarial networks for extreme learned image compression. In *CVPR*, 2019. 3

[3] Nasir Ahmed, T. Natarajan, and Kamisetty R Rao. Discrete cosine transform. *IEEE Transactions on Computers*, 1974. 5

[4] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICLR*, 2018. 2, 7

[5] Thomas Boutell and T Lane. Png (portable network graphics) specification version 1.0. *Network Working Group*, pages 1–102, 1997. 3

[6] Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *ACM Workshop on Artificial Intelligence and Security*, pages 3–14. ACM, 2017. 3

[7] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE S&P*, 2017. 1, 6, 7

[8] Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. Ead: elastic-net attacks to deep neural networks via adversarial examples. In *AAAI*, 2018. 3

[9] Ali Dabouei, Sobhan Soleymani, Fariborz Taherkhani, Jeremy Dawson, and Nasser Nasrabadi. Smoothfool: An efficient framework for computing smooth adversarial perturbations. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2665–2674, 2020. 3

[10] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Li Chen, Michael E Kounavis, and Duen Horng Chau. Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression. *arXiv preprint arXiv:1705.02900*, 2017. 3

[11] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Siwei Li, Li Chen, Michael E Kounavis, and Duen Horng Chau. Shield: Fast, practical defense and vaccination for deep learning using jpeg compression. In *ACM SIGKDD*, 2018. 3, 7

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 2, 6

[13] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *CVPR*, 2018. 7

[14] Ranjie Duan, Xingjun Ma, Yisen Wang, James Bailey, A. K. Qin, and Yun Yang. Adversarial camouflage: Hiding physical-world attacks with natural styles. *In CVPR*, 2020. 3

[15] Ranjie Duan, Xiaofeng Mao, A Kai Qin, Yuefeng Chen, Shaokai Ye, Yuan He, and Yun Yang. Adversarial laser beam: Effective physical-world attack to dnns in a blink. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16062–16071, 2021. 3

[16] Gintare Karolina Dziugaite, Zoubin Ghahramani, and Daniel M Roy. A study of the effect of jpg compression on adversarial images. *arXiv preprint arXiv:1608.00853*, 2016. 3

[17] Logan Engstrom, Andrew Ilyas, Hadi Salman, Shibani Santurkar, and Dimitris Tsipras. Robustness (python library), 2019. 6

[18] 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 *CVPR*, 2018. 3

[19] Ruihao Gong, Xianglong Liu, Shenghu Jiang, Tianxiang Li, Peng Hu, Jiazhen Lin, Fengwei Yu, and Junjie Yan. Differentiable soft quantization: Bridging full-precision and low-bit neural networks. In *CVPR*, 2019. 5

[20] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2014. 1, 3, 6, 7

[21] Chuan Guo, Jared S Frank, and Kilian Q Weinberger. Low frequency adversarial perturbation. In *Uncertainty in Artificial Intelligence*, 2020. 3

[22] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering adversarial images using input transformations. In *ICLR*, 2018. 3, 7

[23] Qing Guo, Felix Juefei-Xu, Xiaofei Xie, Lei Ma, Jian Wang, Bing Yu, Wei Feng, and Yang Liu. Watch out! motion is blurring the vision of your deep neural networks. *In NeurIPS*, 33, 2020. 3

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 6

[25] Paul Heckbert. Color image quantization for frame buffer display. *ACM Siggraph Computer Graphics*, pages 297–307, 1982. 2

[26] Hossein Hosseini and Radha Poovendran. Semantic adversarial examples. In *CVPR Workshop*, 2018. 3

[27] Andrew Ilyas, Shibani Santurkar, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *In NeurIPS*, 2019. 1

[28] Nick Johnston, Damien Vincent, David Minnen, Michele Covell, Saurabh Singh, Troy Chinen, Sung Jin Hwang, Joel Shor, and George Toderici. Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks. In *CVPR*, 2018. 3

[29] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *ICLR*, 2016. 3

[30] Mu Li, Wangmeng Zuo, Shuhang Gu, Debin Zhao, and David Zhang. Learning convolutional networks for content-weighted image compression. In *CVPR*, 2018. 3

[31] Hsueh-Ti Derek Liu, Michael Tao, Chun-Liang Li, Derek Nowrouzezahrai, and Alec Jacobson. Beyond pixel norm-balls: Parametric adversaries using an analytically differentiable renderer. In *ICLR*, 2018. 3

[32] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018. 1, 2, 3, 6, 7

[33] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *CVPR*, 2016. 6, 7

[34] Michael T Orchard, Charles A Bouman, et al. Color quantization of images. *IEEE Transactions on Signal Processing*, pages 2677–2690, 1991. 2

[35] Aaditya Prakash, Nick Moran, Solomon Garber, Antonella DiLillo, and James Storer. Deflecting adversarial attacks with pixel deflection. In *CVPR*, 2018. 2, 3, 6, 7

[36] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, pages 618–626, 2017. 8

[37] Ali Shahin Shamsabadi, Ricardo Sanchez-Matilla, and Andrea Cavallaro. Colorfool: Semantic adversarial colorization. In *CVPR*, 2020. 3

[38] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *CCS*, 2016. 3

[39] Yash Sharma, Gavin Weiguang Ding, and Marcus Brubaker. On the effectiveness of low frequency perturbations. *arXiv preprint arXiv:1903.00073*, 2019. 3

[40] Richard Shin and Dawn Song. Jpeg-resistant adversarial images. In *NeurIPS Workshop*, 2017. 6, 7, 8

[41] Athanassios Skodras, Charilaos Christopoulos, and Touradj Ebrahimi. The jpeg 2000 still image compression standard. *IEEE Signal Processing Magazine*, 2001. 3, 4

[42] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2013. 1, 3

[43] George Toderici, Damien Vincent, Nick Johnston, Sung Jin Hwang, David Minnen, Joel Shor, and Michele Covell. Full resolution image compression with recurrent neural networks. In *CVPR*, 2017. 3

[44] Gregory K Wallace. The jpeg still picture compression standard. *IEEE Transactions on Consumer Electronics*, 1992. 2, 3, 4

[45] Rey Reza Wiyatno and Anqi Xu. Physical adversarial textures that fool visual object tracking. In *ICCV*, 2019. 3

[46] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *NDSS*, 2017. 2, 6, 7

[47] Min Zeng, Yisen Wang, and Yuan Luo. Dirichlet latent variable hierarchical recurrent encoder-decoder in dialogue generation. In *EMNLP*, 2019. 1

[48] Xiaohui Zeng, Chenxi Liu, Yu-Siang Wang, Weichao Qiu, Lingxi Xie, Yu-Wing Tai, Chi-Keung Tang, and Alan L Yuille. Adversarial attacks beyond the image space. In *CVPR*, 2019. 3

[49] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018. 6

[50] Zhengli Zhao, Dheeru Dua, and Sameer Singh. Generating natural adversarial examples. In *International Conference on Learning Representations*, 2018. 3

[51] Zhengyu Zhao, Zhuoran Liu, and Martha Larson. Towards large yet imperceptible adversarial image perturbations with perceptual color distance. In *CVPR*, 2020. 3