Adversarial Camouflage: Hiding Physical-World Attacks with Natural Styles

Ranjie Duan[1] Xingjun Ma[2†] Yisen Wang[3] James Bailey[2] A. K. Qin[1] Yun Yang[1]

[1]Swinburne University of Technology   [2]The University of Melbourne   [3]Shanghai Jiao Tong University

Abstract

Deep neural networks (DNNs) are known to be vulnerable to adversarial examples. Existing works have mostly focused on either digital adversarial examples created via small and imperceptible perturbations, or physical-world adversarial examples created with large and less realistic distortions that are easily identified by human observers. In this paper, we propose a novel approach, called Adversarial Camouflage (AdvCam), to craft and camouflage physical-world adversarial examples into natural styles that appear legitimate to human observers. Specifically, AdvCam transfers large adversarial perturbations into customized styles, which are then "hidden" on-target object or off-target background. Experimental evaluation shows that, in both digital and physical-world scenarios, adversarial examples crafted by AdvCam are well camouflaged and highly stealthy, while remaining effective in fooling state-of-the-art DNN image classifiers. Hence, AdvCam is a flexible approach that can help craft stealthy attacks to evaluate the robustness of DNNs. AdvCam can also be used to protect private information from being detected by deep learning systems. code

1. Introduction

Deep neural networks (DNNs) are a family of powerful models that have been widely used in various AI systems, and they have achieved a great success across many applications, such as image classification [13], speech recognition [26], natural language processing [33], and autonomous driving [6]. However, DNNs are known to be vulnerable to adversarial examples (or attacks) that are crafted by adding carefully designed perturbations on normal examples [25, 12, 20, 2, 27, 28]. This raises serious concerns for security-critical applications [23, 8, 10, 15, 21]. For example, as shown in Figure 1, the addition of a carefully crafted perturbation that resembles stain, snow and discoloration on the surface of a stop sign. A self-driving car equipped with state-of-the-art classifier detects the modified stop sign as other objects with very high confidence (we will examine how this perturbation was created later).
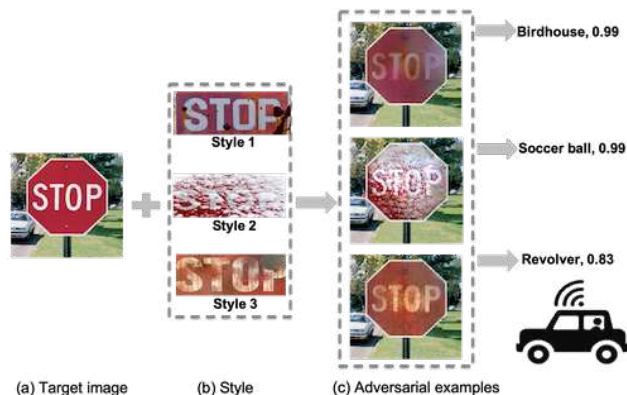

Figure 1: Camouflaged adversarial examples crafted by proposed *AdvCam* attack. Given a target image in (a), an adversary can choose different camouflage styles from (b) to craft adversarial examples in (c) that appear naturally occurring, yet can fool a DNN classifier to make incorrect predictions with high confidence.

Adversarial attacks can be applied in two different settings: 1) **digital setting**, where the attacker can feed input digital images directly into the DNN classifier; and 2) **physical-world setting**, where the DNN classifier only accepts inputs from a camera and the attacker can only present adversarial images to the camera. There are three properties which may be used to characterize adversarial attack: 1) **adversarial strength**, which represents the ability to fool DNNs; 2) **adversarial stealthiness**, which is about whether the adversarial perturbations can be detected by human observers; and 3) **camouflage flexibility**, which is the degree to which the attacker can control the appearance of adversarial image.

Most attacking methods have been developed for the digital setting, such as Projected Gradient Decent (PGD) [22], Carlini and Wagner (CW) attack [4] and adversarial examples crafted using generative adversarial networks (AdvGAN) [30]. For digital attacks, small perturbations are often sufficient. However, physical-world attacks require large or even unrestricted perturbations [17, 1], since small perturbations are too subtle to be captured by cameras in complex physical-world environments. There already exist

---

†Correspondence to: Xingjun Ma (xingjun.ma@unimelb.edu.au)

| (a) RP_2 | (b) AdvCam | (c) AdvPatch | (d) *AdvCam* |

Figure 2: Examples of successful physical-world attacks. *AdvCam* refers to our proposed adversarial camouflage.

several attacking methods that go beyond small perturbations, such as adversarial patch (AdvPatch) [3] and robust physical perturbations (RP_2) [10]. The properties of existing attacks are summarized in Table 1, where ⋆⋆ means better than ⋆. To summarize, stealthiness can be achieved with small perturbations, which are only useful in digital setting. Also, existing attacks require exact perturbation size to achieve stealthiness, while it is difficult to decide a proper perturbation size for both visual imperceptibility and adversarial strength, especially in the physical setting. Besides, the generation process of current methods is difficult to control, e.g., an attacker cannot decide the appearance of adversarial examples. As such, the camouflage flexibility of these methods is rather limited. A flexible (yet strong and stealthy) camouflage mechanism for large perturbations is still an open problem that needs to be addressed.

Table 1: Summary of existing attacks and our *AdvCam*.

Attack	Digital	Physical	Stealthiness	Flexibility
PGD	√	×	⋆⋆	⋆
AdvPatch	×	√	⋆	⋆
RP_2	×	√	⋆	⋆
AdvCam	√	√	⋆⋆	⋆⋆

To address this gap, in this paper, we propose a novel adversarial camouflage approach (*AdvCam*) to craft and camouflage adversarial examples into natural styles using style transfer techniques. The style of an image is an abstract concept that generally refers to its visual appearance such as color and texture, in contrast to its structure information [16]. In *AdvCam*, the camouflage style and attack region can be customized by the attacker according to different attacking scenarios. For example, Figure 1 shows several adversarial traffic signs crafted by our *AdvCam* attack. A quick visual comparison of *AdvCam* to existing physical-world attacks can be found in Figure 2. While all the adversarial examples in Figure 2 attack DNNs successfully, we can see that *AdvCam* is able to generate adversaial perturbation on stop sign with natural stains compared to artificial graffiti created by RP_2, or a camouflaged product label compared to a patch with obtrusive pattern generated by AdvPatch. Our proposed *AdvCam* is capable of generating highly stealthy adversarial examples, which are robust to various physical-world conditions at the same time.

In summary, *AdvCam* is not a perturbation-restricted attack and therefore is not inherently subjected to the limited amount of perturbation which is typically required in existing perturbation-restricted techniques. We define a flexible mechanism that induces perturbation appearing in natural-looking, which is a totally different paradigm from previous attacks. It is such an intrinsic difference of the working principles that makes *AdvCam* to produce more realistic images than existing methods.

Our key contributions in this paper are:

- We propose a flexible adversarial camouflage approach, *AdvCam*, to craft and camouflage adversarial examples.

- *AdvCam* allows the generation of large perturbations, customizable attack regions and camouflage styles. It is very flexible and useful for vulnerability evaluation of DNNs against large perturbations for physical-world attacks.

- Experiments on both digital and physical-world scenarios show that adversarial examples camouflaged by *AdvCam* are highly stealthy, while remaining effective in fooling state-of-the-art DNN image classifiers.

2. Related Work

2.1. Adversarial Attack

Adversarial attack is to generate adversarial examples by maximizing the classification error of the target model (the model to attack) [25]. There are targeted and untargeted attacks. Targeted attack is to fool the network to misclassify the adversarial example into the class that attacker expects, while untargeted attack is to fool the network to misclassify the adversarial example into any incorrect classes [12]. Adversarial attacks can be applied either in a digital setting directly on the target model, or in a physical-world setting, where recaptured photos of adversarial examples are fed to the target model [17].

2.1.1 Digital attacks

Adversarial examples can be crafted by one or more steps of perturbation following the direction of adversarial gradients. This includes the classic Fast Gradient Sign Method (FGSM) [12], the Basic Iterative Method (BIM) [17], the strongest first-order method Projected Gradient Decent (PGD) [22], and the Skip Gradient Method (SGD) [29] for transferable attacks. These attacks can either be targeted or untargeted, and their perturbations are bounded by a small norm-ball $\|\cdot\|_p \leq \epsilon$ with L_2 and L_∞ being the most commonly used norms. Optimization-based attacks, such as Carlini and Wagner (CW) attack [4] and elastic-net (EAD) attack [7], directly minimize the perturbations as part of the adversarial loss.

There also exist several unrestricted adversarial attacks. These attacks search for modifications on substitutable components (attributes) of an image such as color [14], texture and physical parameters [34, 18] while preserving critical components of images. However, these attacks either produce large unnatural distortions or rely on training data that has semantic annotations. Moreover, these attacks cannot generate complex adversarial patterns and thus are quite limited for complicated real scenarios. Adversarial examples can also be generated by generative adversarial networks (GANs) [30, 24]. However, it is difficult to craft targeted attacks for a given test image with GANs, since the generation process is hard to control.

Most existing attacks achieve stealthiness by either crafting small perturbations or modifying semantic attributes of the target image. However, a flexible camouflage mechanism that can effectively hide adversarial examples with natural styles is still missing from the literature. In this paper, we address this gap by proposing one such approach.

2.1.2 Physical-world attacks

A study has shown that, by printing and recapturing using a cell-phone camera, digital adversarial examples can still be effective [17]. However, follow-up works have found that such attacks are not easy to realize under physical-world conditions, due to viewpoint shifts, camera noise, and other natural transformations [1]. Thus, strong physical-world attacks require large perturbations, and specific adaptations over a distribution of transformations including lighting, rotation, perspective projection etc. The AdvPatch attack allows large perturbations and is immune to scaling or rotation, thus being directly applicable as a physical-world attack [3]. Adversarial stickers and graffiti have also been used to attack such as traffic sign classifiers and ImageNet classifiers in physical-world scenarios [10]. Other physical-world attacks include adversarial eye-glass frames [23], vehicles [35], or t-shirts [32] that can fool face recognition systems or object detectors. All these physical-world attacks generate large perturbations to increase adversarial strength, which inevitably results in large and unrealistic distortions. This greatly reduces their stealthiness, as shown in Figure 2.

2.2. Neural Style Transfer

Neural style transfer evolves from the problem of texture transfer, for which the goal is to transfer the texture of a source image to a target image while preserving the structural information of the target image. Traditional texture transfer methods mainly focus on non-parametric algorithms that resample pixels from the source texture [9]. However, these methods suffer from the fundamental limitation of pixel replacement, i.e., they cannot handle complex styles. Neural style transfer demonstrates remarkable

results for image stylization [11]. In it, the content and style information of an image can be separated from its feature representations learned by a convolutional neural network (CNN). Then, the style information of the style image can be recombined into the target image to achieve style transfer. This technique has attracted several follow-up works for different aspects of improvement [5, 19]. In this paper, we will exploit these techniques for the camouflage of adversarial examples.

3. Our Adversarial Camouflage Approach

In this section, we first give an overview of the adversarial attack problem and our proposed camouflage approach. We then introduce the loss functions used by our proposed approach, and adaptations for physical world conditions.

3.1. Overview

Given a test image $x \in \mathbb{R}^m$ with class label y, a DNN classifier $F : \mathbb{R}^m \to \{1, \cdots, k\}$ mapping image pixels to a discrete label set, and a target class $y_{adv} \neq y$, adversarial attack is to find an adversarial example x' for target image x by solving the following optimization problem:

$$\begin{aligned} \text{minimize} \quad & \mathcal{D}(x, x') + \lambda \cdot \mathcal{L}_{adv}(x') \\ \text{such that} \quad & x' \in [0, 255]^m, \end{aligned} \quad (1)$$

where $\mathcal{D}(x, x')$ is a distance metric that defines the stealthiness property of the adversarial example, $\mathcal{L}_{adv}$ is the adversarial loss, $[0, 255]$ indicates the valid pixel values, λ is a parameter that adjusts the adversarial strength. Note that there is a trade-off between stealthiness and adversarial strength. In the whole experiments, we fix all other parameters as constants [19], and only adjust adversarial strength parameter λ.

Our goal is to develop a mechanism that crafts and camouflages adversarial examples with large perturbations into customized styles, and either attack area or style can be defined by attacker flexibly. We use style transfer techniques to achieve the goal of camouflage and adversarial attack techniques to achieve adversarial strength. The final loss is a combination of an adversarial loss $\mathcal{L}_{adv}$ for adversarial strength, a style loss $\mathcal{L}_s$ for style generation, a content loss $\mathcal{L}_c$ to preserve the content of the source image and a smoothness loss $\mathcal{L}_m$ to generate locally smooth regions. We denote this final loss as the *adversarial camouflage loss*:

$$\mathcal{L} = (\mathcal{L}_s + \mathcal{L}_c + \mathcal{L}_m) + \lambda \cdot \mathcal{L}_{adv}, \quad (2)$$

where the three loss functions in brackets together serve the purpose of camouflage. The overview of our approach is illustrated in Figure 3. An attacker defines target image, target attack region and expected target style. Our proposed *AdvCam* then generates adversarial perturbation with expected style on the expected area as shown on the right of
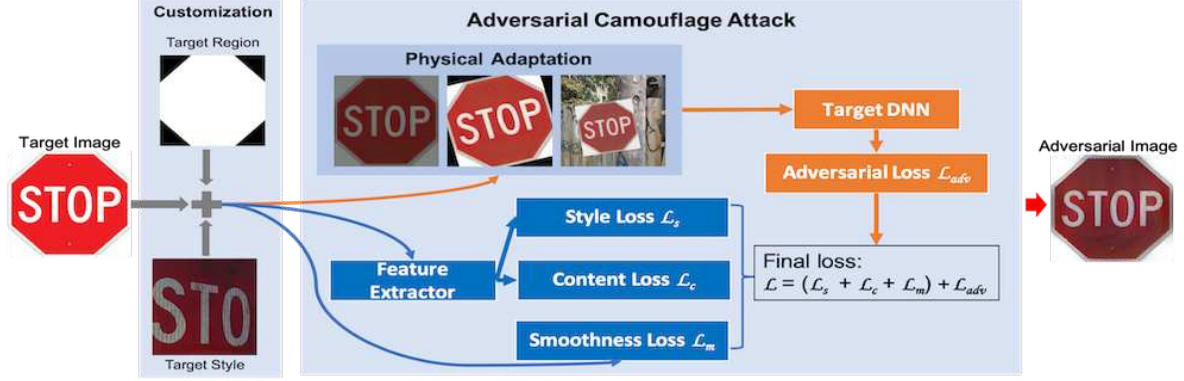
Figure 3: Overview of the proposed approach.

Figure 3. To make adverarial example robust to various environment conditions, including lighting, rotation, etc., we add an extra physical adaptation training for generated x' in each step.

3.2. Adversarial Camouflage Loss

3.2.1 Style loss

For traditional attacks, the stealthiness metric is defined by $\mathcal{D}(x, x') = \|x - x'\|_p$, where $\|\cdot\|_p$ is the L_p norm and L_2 and L_∞ are typically used. This is to constrain the perturbations to be "small". For our proposed camouflage, the stealthiness is defined by a style metric between adversarial example x' and a style reference image x^s. The style distance between two images can be defined by their differences in style representations:

$$\mathcal{D}_s = \sum_{l \in \mathcal{S}_l} \left\| \mathcal{G}(\widetilde{F}_l(x^s)) - \mathcal{G}(\widetilde{F}_l(x')) \right\|_2^2, \quad (3)$$

where $\widetilde{F}$ is a feature extractor (such as a public DNN model) that can be different from the target model, and $\mathcal{G}$ is the Gram matrix of deep features extracted at a set of style layers of $\widetilde{F}$ [11]. As different styles can be learned at different layers, we use all convolutional layers of the network as the style layers. To generate stylized perturbation in expected area of the target image, we denote the masks that define the attack and non-attack regions by M and $\overline{M}$ respectively. After each step in generation process, we mask the adversarial image x' by M to make only attack region modifiable with non-attack area unchanged (x masked by $\overline{M}$).

3.2.2 Content loss

The above style loss can craft an adversarial image in the reference style, however, the content of the adversarial image may appear very different to that of the original image. The content of the original image can be preserved by a content preservation loss:

$$\mathcal{L}_c = \sum_{l \in \mathcal{C}_l} \left\| \widetilde{F}_l(x) - \widetilde{F}_l(x') \right\|_2^2, \quad (4)$$

where $\mathcal{C}_l$ represents the set of content layers used for extracting content representations. This is to ensure that the adversarial image has very similar content to the original image in the deep representation space. We use deeper layers of the feature extractor network as the content layers.

Note that the content loss is optional when the attack only occurs in a small region that does not contain any particular content. However, if the attack region contains semantic content, the content loss can help reduce the semantic difference between the adversarial image and its original version.

3.2.3 Smoothness loss

The smoothness of the adversarial image can be improved by reducing the variations between adjacent pixels. For an adversarial image x', the smoothness loss is defined as:

$$\mathcal{L}_m = \sum ((x'_{i,j} - x_{i+1,j})^2 + (x'_{i,j} - x_{i,j+1})^2)^{\frac{1}{2}}, \quad (5)$$

where $x'_{i,j}$ is the pixel at coordinate (i, j) of image x'. Intuitively, this will encourage the image to have low-variance (e.g. smooth) local patches. We note that the smoothness loss is limited in improving stealthiness if the surface of both the target image and the style image are already smooth. But we still recommend adding it in physical setting, as Sharif et al. pointed out [23], the smoothness term is useful for improving the robustness of adversarial examples in physical environment.

3.2.4 Adversarial loss

For the adversarial loss, we use the following cross-entropy loss:

$$\mathcal{L}_{adv} = \begin{cases} \log(p_{y_{adv}}(x')), & \text{for targeted attack} \\ -\log(p_y(x')), & \text{for untargeted attack,} \end{cases} \quad (6)$$

where $p_{y_{adv}}()$ is the probability output (softmax on logits) of the target model F with respect to class y_{adv}. We note that the proposed camouflage attack is not restricted to the particular form of the adversarial loss, and can be used in combination with existing attacking methods.

3.3. Adaptation for Physical-world Conditions

To make adversaries generated by *AdvCam* physically realizable, we model physical conditions in the process of generating camouflaged adversaries. As the physical-world environment often involves condition fluctuations such as viewpoint shifts, camera noise, and other natural transformations [1], we use a series of adaptations to accommodate such varying conditions. In particular, we adapt a technique similar to Expectation Over Transformation (EOT) [1] but without expectation. In Xie's work [31], they also adapted EOT to improve the transferability of adversarial examples. However, we aim to improve the adaptation of adversarial examples under various physical conditions. Thus we consider the transformations for simulation of physical-world condition fluctuations, including rotation, scale resize, color shift (to simulate lightening change) and random background.

$$\min_{x'} \left((\mathcal{L}_s + \mathcal{L}_c + \mathcal{L}_m) + \max_{T \in \mathcal{T}} \lambda \cdot \mathcal{L}_{adv}(o + T(x')) \right), \quad (7)$$

where o represents a random background image that we sample in the physical world, and T represents a random transformation of rotation, resize and color shift.

In principle, "vision" is the main sense of perception for both human observers and DNN models. By adjusting the style reference image x^s according to the original image x and the background image o, the proposed camouflage attack can craft highly stealthy adversarial examples that can deceive both human observers and DNN models.

4. Experimental Evaluation

In this section, we first outline the experimental setup. Then we analyze our *AdvCam* attack via an ablation study. Afterwords, we evaluate camouflage performance of AdvCam by human perception study for digital attacks, and also present several adversarial examples of high stealthiness crafted by *AdvCam*. We further perform physical-world attacks at last.

4.1. Experimental Setup

4.1.1 Baseline attacks

we compare our *AdvCam* attack with two existing representative methods: PGD [22] and adversarial patch (AdvPatch) [3]. PGD represents the digital attacks, which is the strongest first-order attack. And AdvPatch represents the unrestricted attacks that can directly applied to the physical-world setting. Some other physical-world attacks such as RP_2 require case-by-case manual design, thus are limited for mass production. We compare the methods in terms of attack success rate and visual effect. For *AdvCam* attack, we use the same layers of the network as used in [19] to extract style and content (where necessary) representations.

4.1.2 Threat model

We test both targeted and untargeted attacks in both digital and physical-world settings. The threat model adopts a gray-box setting: the source and target networks are both VGG-19 networks but were trained separately on ImageNet. For a physical-world test, we first use a Google Pixel2 smartphone to take a photo of the target object, then craft an adversarial image in the digital setting attacking the source network, after which we print out the adversarial image to replace or place it on the target object, then we re-photo the object using the same smartphone from various viewing angles and distances. The physical-word attack success rate is measured by the prediction accuracy of the target network on the re-photoed adversarial images.

4.2. Ablation Study

Here, we conduct a series of experiments on ImageNet to analyze the following aspects of our *AdvCam* attack: 1) shape and location of camouflage region, 2) camouflage losses (e.g. style loss, content loss and smoothness loss), and 3) adversarial strength parameter λ and region size.

4.2.1 Camouflage region: shape and location

Here, we show how the selection of camouflage region in terms of shape and location impacts the adversarial strength of the crafted adversarial example. Given a selected attacking region with shape and size, we increase strength parameter λ from 1000 to 10000 with interval 1000 until the attack succeeds. The range is selected according to extensive experiments, [1000, 10000] is an effective range to find adversary with both high adversarial strength and stealthiness. Figure 4 shows camouflaged adversarial examples crafted with different regions. We find that either the camouflage region is at the center or away from the center of the target object, attacks can succeed with high confidence. We will show in the last part of this subsection that attacks can be camouflaged into an area that is even off the target object, secretly hidden in the background.

Figure 4: Ablation of region shape and size, on two targeted attacks: "backpack" → "tank" (top row) and "poncho" → "traffic light". (a): original clean image with intended style (bottom left corner). (b) - (e): left: selected attack region, right: crafted camouflage adversarial example with top-1 prediction ("predicted class, confidence") given at the bottom of the image.

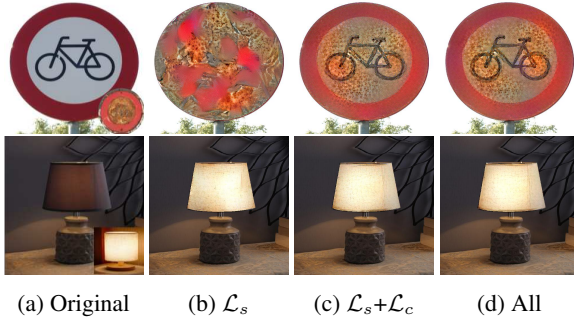(a) Original   (b) $\mathcal{L}_s$ (c) $\mathcal{L}_s + \mathcal{L}_c$ (d) All

Figure 5: Ablation of the 3 camouflage losses: (a): original images with intended camouflage style at the bottom right corner; (b) - (d): camouflaged adversarial examples using different loss functions.

4.2.2 Camouflage losses ($\mathcal{L}_s$, $\mathcal{L}_c$, $\mathcal{L}_m$)

Figure 5 illustrates three groups of adversarial examples camouflaged with or without the two optional enhancements (content preservation $\mathcal{L}_c$ and smoothness enhancement $\mathcal{L}_m$). When incorporating one enhancement, its loss function is directly added to the final object by following Eq. (2) (λ in $\mathcal{L}_{adv}$ was set to 2000). As can be observed, the content preservation can help preserve the original content as in the "traffic sign" example (third column), while smoothness enhancement can help produce a smooth object surface. These enhancements are optional because they improve the visual appearance only slightly for some adversarial examples, for example, content preservation in the "table lamp" example (third row) or smoothness enhancement in all examples.

4.2.3 Adversarial strength (λ) and region size

We craft both targeted and untargeted camouflage attacks on randomly selected 2000 ImageNet test images from 50 categories with varying $\lambda \in [1000, 10000]$. For targeted at-
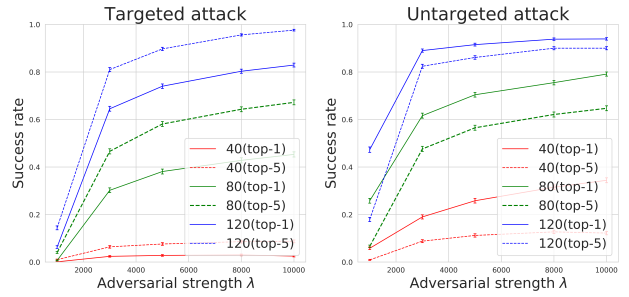

Figure 6: Ablation of adversarial strength λ and region size: success rate of untargeted (left) and targeted attack (right).

tack, the target class is randomly selected and is different to the true class. To also test the influence of λ under different attack regions, here we also vary the size of the region from 40*40 to 120*120. Figure 6 illustrates the top-1/5 success rates, which are measured by whether the target class is in the top-1/5 classes. As shown in the figure, when the region is fixed, larger adversarial strength λ can increase success rate by up to 20%; and when λ is fixed, larger attack region can improve success rate by up to 40%. Compared to targeted attacks, untargeted attacks are much easier to succeed. Between top-1 and top-5 success rates, top-1 targeted attacks are more difficult to achieve (solid lines are lower than dashed lines). The standard errors with respect to different adversarial strengths and region sizes are between 0.07% and 1.13%. Note that in these experiments, the camouflage styles and locations of attack region are selected randomly, which may decrease the camouflage effect and success rate. However, this does not affect the conclusion that larger λ and region size can help craft stronger attacks.

4.3. Digital Attacks

4.3.1 Attack setting

We randomly select 150 clean images from 5 categories of ImageNet ILSVRC2012 test set. We then apply the

three methods (PGD, AdvPatch and our *AdvCam*) to craft a targeted adversarial example for each clean image. The pairs of selected source and target classes are: "ashcan" → "robin", "backpack" → "water jug", "cannon" → "folklift", "jersey" → "horizontal bar", "mailbox" → "entertainment center". For PGD and *AdvCam*, we attack the main object region obtained via manual selection, while for Adv-Patch, we further select a circular attack area inside of the object region. For PGD, we use maximum perturbation $\epsilon = 16/255$ (denoted as PGD-16). For *AdvCam*, we randomly select a second image from the same category as the style image, and gradually increase λ from 1000 to 10000 until an adversarial example is found. For fair comparison, we filter out the failed adversarial examples. Finally, we collect 132, 101, 122 adversarial examples for PGD, AdvPatch and *AdvCam* respectively. Figure 7 shows a few crafted adversarial examples by the three methods that we used to perform human perception study.

(a) Original	(b) PGD-16	(c) AdvPatch	(d) *AdvCam*

Figure 7: The original and adversarial images crafted by PGD-16, AdvPatch and our *AdvCam* attack.

4.3.2 Human perception study results

We set up a human perception study on Amazon Mechanical Turk (AMT) to ask human evaluators to choose whether a shown image is "natural and realistic" or "not natural or realistic". To simulate adversarial examples in a real world scenario, we present users with three types of adversarial images in random order and individually rather than in pairs. Finally, we collected 1953 selections from 130 participants. AdvPatch was chosen as "natural and realistic" $19.0 \pm 1.68\%$ of the time, PGD was chosen $77.3 \pm 1.53\%$ of the time, while our *AdvCam* was chosen $80.7 \pm 1.53\%$ of the time. We summarize these statistics as stealthiness scores for the three methods and show in Figure 8. This confirms that our proposed *AdvCam* attack is capable of crafting adversarial examples that are as stealthy as the small perturbation PGD-16 method, although the perturbations of *AdvCam* attacks are unrestricted in size.
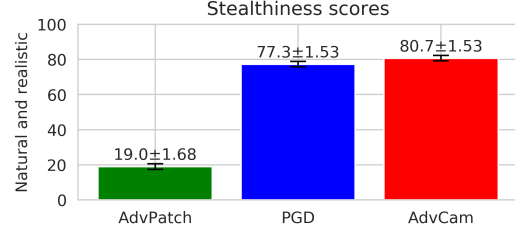

Figure 8: Stealthiness of AdvPatch, PGD-16 and *AdvCam*.

4.3.3 Customized examples

Here, we show how *AdvCam* can craft extremely stealthy camouflages, especially off-target ones. Figure 9 illustrates a few such examples. The first-row shows on-target camouflage examples, and the second-row shows off-target camouflages, which are crafted by attacking a carefully chosen background area. For on-target ones, *AdvCam* generates natural and stealthy perturbation on the surface of the target objects. For off-target ones, in the first off-target example (left two images in the second row), we hide the attack into the price tag to fool a VGG-19 classifier to misclassify a revolver into a toilet tissue. In the second example (middle two images in the second row), *AdvCam* successfully camouflages a blenheim spaniel into a bearskin by adding flowers in the background. In the third example (right two images in the second row), we camouflage the attack into the wall posters in the background which causes a minivan parked in front of the wall to be misrecognized as a traffic light. These examples not only demonstrate the stealthiness and flexibility of our *AdvCam* attack, but also indicate that threats to deep learning systems are ubiquitous, and in many cases, may hardly be noticeable even to human observers.

4.4. Physical-world Attacks

We further design three physical-world attacking scenarios to test the camouflage power of our *AdvCam* attack. We also perform AdvPatch and PGD attacks for comparison. For PGD attack, we test different ϵ in (16/255, 32/255, 64/255, 128/255) and show successful adversarial examples with the least ϵ. We print out the adversarial patterns on a A3 or A4 paper, then take 20 photos at various viewing angles and distances.

The fist scenario is to camouflage a wild pattern into a street sign, which could cause problems for self-driving cars. The top row in Figure 10 illustrates some successful patterns crafted by PGD-128 ($\epsilon = 128/255$), AdvPatch and *AdvCam*. As can be seen, the attack is perfectly camouflaged by *AdvCam* into the texture of the tree. Although PGD is highly stealthy in digital setting, it requires large perturbation ($\epsilon = 128/255$) in physical environment. Thus the adversarial pattern is much less stealthy than *AdvCam*, same as AdvPatch. The second scenario is to protect the identity of a person wearing a jersey. We simulate such a

Figure 9: Camouflaged adversarial images crafted by our *AdvCam* attack and their original versions.

| (a) AdvPatch | (b) PGD-128 | (c) *AdvCam* |

Figure 10: *Top*: Adversarial wood texture recognized as street sign. *Bottom*: Adversarial logo on t-shirt.

Figure 11: Adversarial traffic sign with 3 styles of stains.

scenario by attacking the jersey using a camouflaged fashion logo "pikachu" (see bottom row in Figure 10). All the three attacks perform the "jersey" to "Irish terrier" attack. Note that PGD failed the attack even with the largest perturbation tested. This shows the high flexibility of *AdvCam* with customized camouflage styles, providing a flexible creation of stealthiness satisfying various attacking scenarios.

We also perform a "street sign" to "barbershop" attack using *AdvCam* with three different natural styles (see Figure

11). The patterns of *AdvCam* are smooth and natural that can hardly be detected by human observers, but deceive the classifier with high confidence successfully. To summarize, with high stealthiness of adversaries generated by *AdvCam*, it poses ubiquitous threats for current DNNs-based systems. Thus *AdvCam* can be a useful tool to evaluate the robustness of DNNs employed in the physical-world.

5. Conclusion and Future Work

In this paper, we have investigated the stealthiness of adversarial examples, and propose a novel approach called adversarial camouflage (*AdvCam*), which combines neural style transfer and adversarial attack techniques, to craft and camouflage adversarial examples into stealthy natural-looking styles. *AdvCam* is a flexible approach that can help craft stealthy attacks for robustness evaluation of DNN models. Apart from the view of attack, the proposed *AdvCam* can be a meaningful camouflage technique to protect objects or human being detected by both human observers and DNN based equipments.

The proposed *AdvCam* currently still requires the attacker to manually specify the attack region and target style, where we plan to explore semantic segmentation techniques to automatically achieve this in our future work. Also, we will explore to apply *AdvCam* on other computer vision tasks including object detection and segmentation. Moreover, effective defense strategies against camouflaged attacks will be another crucial and promising direction.

Acknowledgement

References

[1] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *ICLR*, 2017. 1, 3, 5

[2] Yang Bai, Yan Feng, Yisen Wang, Tao Dai, Shu-Tao Xia, and Yong Jiang. Hilbert-based generative defense for adversarial examples. In *ICCV*, 2019. 1

[3] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. In *NIPS Workshop*, 2017. 2, 3, 5

[4] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE S&P*, 2017. 1, 2

[5] Alex J Champandard. Semantic style transfer and turning two-bit doodles into fine artworks. In *ICLR*, 2016. 3

[6] Chenyi Chen, Ari Seff, Alain Kornhauser, and Jianxiong Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In *ICCV*, 2015. 1

[7] 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. 2

[8] Yinpeng Dong, Hang Su, Baoyuan Wu, Zhifeng Li, Wei Liu, Tong Zhang, and Jun Zhu. Efficient decision-based black-box adversarial attacks on face recognition. In *CVPR*, pages 7714–7722, 2019. 1

[9] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In *PACMCGIT*, 2001. 3

[10] Ivan Evtimov, Kevin Eykholt, Earlence Fernandes, Tadayoshi Kohno, Bo Li, Atul Prakash, Amir Rahmati, and Dawn Song. Robust physical-world attacks on deep learning models. In *CVPR*, 2018. 1, 2, 3

[11] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, 2016. 3, 4

[12] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2014. 1, 2

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

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

[15] Linxi Jiang, Xingjun Ma, Shaoxiang Chen, James Bailey, and Yu-Gang Jiang. Black-box adversarial attacks on video recognition models. In *ACM MM*, 2019. 1

[16] Sergey Karayev, Matthew Trentacoste, Helen Han, Aseem Agarwala, Trevor Darrell, Aaron Hertzmann, and Holger Winnemoeller. Recognizing image style. *arXiv preprint arXiv:1311.3715*, 2013. 2

[17] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *ICLR*, 2016. 1, 2, 3

[18] 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

[19] Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. Deep photo style transfer. In *CVPR*, 2017. 3, 5

[20] Xingjun Ma, Bo Li, Yisen Wang, Sarah M Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E Houle, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. In *ICLR*, 2018. 1

[21] Xingjun Ma, Yuhao Niu, Lin Gu, Yisen Wang, Yitian Zhao, James Bailey, and Feng Lu. Understanding adversarial attacks on deep learning based medical image analysis systems. In *arXiv:1907.10456*, 2019. 1

[22] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018. 1, 2, 5

[23] 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. 1, 3, 4

[24] Yang Song, Rui Shu, Nate Kushman, and Stefano Ermon. Constructing unrestricted adversarial examples with generative models. In *NIPS*, 2018. 3

[25] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2013. 1, 2

[26] Yisen Wang, Xuejiao Deng, Songbai Pu, and Zhiheng Huang. Residual convolutional ctc networks for automatic speech recognition. *arXiv preprint arXiv:1702.07793*, 2017. 1

[27] Yisen Wang, Xingjun Ma, James Bailey, Jinfeng Yi, Bowen Zhou, and Quanquan Gu. On the convergence and robustness of adversarial training. In *ICML*, 2019. 1

[28] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *ICLR*, 2020. 1

[29] Dongxian Wu, Yisen Wang, Shu-Tao Xia, James Bailey, and Xingjun Ma. Skip connections matter: On the transferability of adversarial examples generated with resnets. In *ICLR*, 2020. 2

[30] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks. In *IJCAI*, 2018. 1, 3

[31] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *CVPR*, 2019. 5

[32] Kaidi Xu, Gaoyuan Zhang, Sijia Liu, Quanfu Fan, Mengshu Sun, Hongge Chen, Pin-Yu Chen, Yanzhi Wang, and Xue Lin. Adversarial t-shirt! evading person detectors in a physical world. *arXiv preprint arXiv:1910.11099*, 2019. 3

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

[34] 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

[35] Yang Zhang, Hassan Foroosh, Philip David, and Boqing Gong. Camou: Learning physical vehicle camouflages to adversarially attack detectors in the wild. In *ICLR*, 2019. 3