information

MDPI

Article

AdvRain: Adversarial Raindrops to Attack Camera-Based Smart Vision Systems

Amira Guesmi *, Muhammad Abdullah Hanif and Muhammad Shafique

eBrain Lab, Division of Engineering, New York University Abu Dhabi (NYUAD), Abu Dhabi 129188, United Arab Emirates
* Correspondence: ag9321@nyu.edu

Abstract: Vision-based perception modules are increasingly deployed in many applications, especially autonomous vehicles and intelligent robots. These modules are being used to acquire information about the surroundings and identify obstacles. Hence, accurate detection and classification are essential to reach appropriate decisions and take appropriate and safe actions at all times. Current studies have demonstrated that "printed adversarial attacks", known as physical adversarial attacks, can successfully mislead perception models such as object detectors and image classifiers. However, most of these physical attacks are based on noticeable and eye-catching patterns for generated perturbations making them identifiable/detectable by the human eye, in-field tests, or in test drives. In this paper, we propose a camera-based inconspicuous adversarial attack (**AdvRain**) capable of fooling camera-based perception systems over all objects of the same class. Unlike mask-based FakeWeather attacks that require access to the underlying computing hardware or image memory, our attack is based on emulating the effects of a natural weather condition (i.e., Raindrops) that can be printed on a translucent sticker, which is externally placed over the lens of a camera whenever an adversary plans to trigger an attack. Note, such perturbations are still inconspicuous in real-world deployments and their presence goes unnoticed due to their association with a natural phenomenon. To accomplish this, we develop an iterative process based on performing a random search aiming to identify critical positions to make sure that the performed transformation is adversarial for a target classifier. Our transformation is based on blurring predefined parts of the captured image corresponding to the areas covered by the raindrop. We achieve a drop in average model accuracy of more than 45% and 40% on VGG19 for ImageNet dataset and Resnet34 for Caltech-101 dataset, respectively, using only 20 raindrops.

Keywords: adversarial machine learning; physical adversarial attack; security; efficiency; perturbations; physical attacks; deep neural networks; DNNs; classification; object detection; camera; autonomous systems; robots; autonomous vehicles; Grad-CAM; random-search

Citation: Guesmi, A.; Hanif, M.A.; Shafique, M. AdvRain: Adversarial Raindrops to Attack Camera-Based Smart Vision Systems. *Information* **2023**, *14*, 634. https://doi.org/10.3390/info14120634

Academic Editor: Alessandra Lumini

Received: 31 July 2023
Revised: 15 September 2023
Accepted: 20 September 2023
Published: 28 November 2023

1. Introduction

The revolutionary emergence of deep learning (DL) has shown a profound impact across diverse sectors, particularly in the realm of autonomous driving [1]. Prominent players in the automotive industry, such as Google, Audi, BMW, and Tesla, are actively harnessing this cutting-edge technology in conjunction with cost-effective cameras to develop autonomous vehicles (AVs). These AVs are equipped with state-of-the-art vision-based perception modules, empowering them to navigate real-life scenarios even under high-pressure circumstances, make informed decisions, and execute safe and appropriate actions.

Consequently, the demand for autonomous vehicles has soared, leading to substantial growth in the AV market. Strategic Market Research (SMR) predicts that the autonomous vehicle market will achieve an astonishing valuation of $196.97 billion by 2030, showcasing an impressive compound annual growth rate (CAGR) of 25.7% (ACMS). The integration of

DL-powered vision-based perception modules has undeniably accelerated the progress of autonomous driving technology, heralding a transformative era in the automotive industry. With the increasing prevalence of AVs, their potential impact on road safety, transportation efficiency, and overall user experience remains a subject of great interest to consumers, researchers, and investors alike.

However, despite the significant advancements in deep learning models, they are not immune to adversarial attacks, which can pose serious threats to their integrity and reliability. Adversarial attacks involve manipulating the input of a deep learning classifier by introducing carefully crafted perturbations, strategically chosen by malicious actors, to force the classifier into producing incorrect outputs. Such vulnerabilities can be exploited by attackers to compromise the security and integrity of the system, potentially endangering the safety of individuals interacting with it. For instance, a malicious actor could add adversarial noise to a stop sign, causing an autonomous vehicle to misclassify it as a speed limit sign [2,3]. This kind of misclassification could lead to dangerous consequences, including accidents and loss of life. Notably, adversarial examples have been shown to be effective in real-world conditions [4]. Even when printed out, an image specifically crafted to be adversarial can retain its adversarial properties under different lighting conditions and orientations.

Therefore, it becomes crucial to understand and mitigate these adversarial attacks to ensure the development of safe and trustworthy intelligent systems. Taking measures to defend against such attacks is imperative for maintaining the reliability and security of deep learning models, particularly in critical applications such as autonomous vehicles, robotics, and other intelligent systems that interact with people.

Adversarial attacks can broadly be categorized into two types: *Digital Attacks* and *Physical Attacks*, each distinguished by its unique form of attack [3–5]. In a *Digital Attack*, the adversary introduces imperceptible perturbations to the digital input image, specifically tailored to deceive a given deep neural network (DNN) model. These perturbations are carefully optimized to remain unnoticed by human observers. During the generation process, the attacker works within a predefined noise budget, ensuring that the perturbations do not exceed a certain magnitude to maintain imperceptibility. In contrast, *Physical Attacks* involve crafting adversarial perturbations that can be translated into the physical world. These physical perturbations are then deployed in the scene captured by the victim DNN model. Unlike digital attacks, physical attacks are not bound by noise magnitude constraints. Instead, they are primarily constrained by location and printability factors, aiming to generate perturbations that can be effectively printed and placed in real-world settings without arousing suspicion.

The primary objective of an adversarial attack and its relevance in real-world scenarios is to remain inconspicuous, appearing common and plausible rather than overtly hostile. Many previous works in developing adversarial patches for image classification have focused mainly on maximizing attack performance and enhancing the strength of adversarial noise. However, this approach often results in conspicuous patches that are easily recognizable by human observers. Another line of research has aimed to improve the stealthiness of the added perturbations by making them blend seamlessly into natural styles that appear legitimate to human observers. Examples include camouflaging the perturbations as color films [6], shadows [7], or laser beams [8], among others.

In Figure 1, we provide a visual comparison of AdvRain with existing physical attacks. While all the adversarial examples in Figure 1 successfully attack deep neural networks (DNNs), AdvRain stands out in its ability to generate adversarial perturbations with natural blurring marks (emulating a similar phenomenon to the actual rain), unlike the conspicuous pattern generated by AdvPatch or the unrealistic patterns generated by FakeWeather [9]. This showcases the effectiveness of AdvRain in creating adversarial perturbations that blend in with the surrounding environment, making them difficult for human observers to detect.

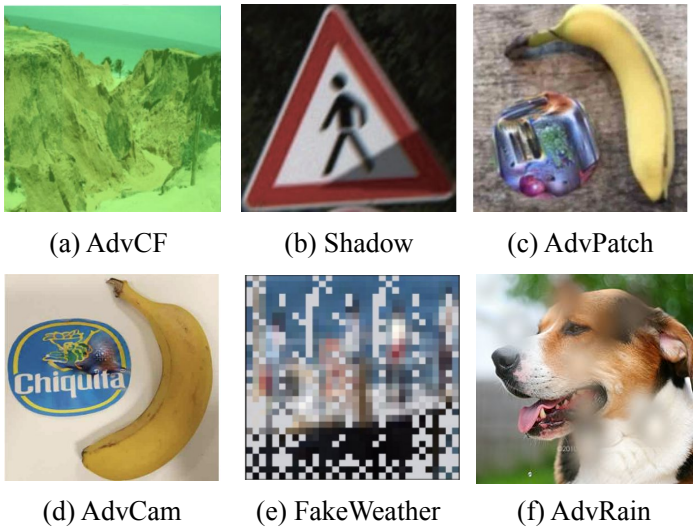

Figure 1. AdvRain vs. existing physical-world attacks: (**a**) AdvCF [7], (**b**) Shadow [6], (**c**) advPatch [10], (**d**) AdvCam [8], (**e**) FakeWeather [9], and (**f**) AdvRain.

In this paper, we present a novel technique aimed at deceiving a given DNN model by introducing a subtle perturbation that causes misclassification of all objects belonging to a specific class. Our approach involves creating an adversarial camera sticker, designed to be attached to the camera's lens. This sticker features a carefully crafted pattern of raindrops, which, when perceived by the camera, leads to misclassification of the captured images (See Figure 2). The patterns created by the raindrops appear as water drops in the camera image, making them inconspicuous to human observers.

Figure 2. Adversarial examples generated by **AdvRain** and their corresponding attack success rate when using only 10 raindrops.

Unlike previous adversarial attacks that typically operate at the pixel-level granularity of the images, our main challenge lies in the limited space of feasible perturbations that can be introduced using this camera sticker model. The physical optics of the camera constrain us, resulting in the production of blurry dots as the primary perturbation. These blurry dots lack the high-frequency patterns commonly found in traditional adversarial attacks. Our research addresses the unique challenges posed by physical adversarial attacks, where the goal is to exploit the limitations and characteristics of the camera's optics to achieve

stealthiness. By developing this technique, we contribute to the understanding of physical adversarial attacks and provide insights into developing robust defense mechanisms for intelligent systems in real-world scenarios.

In this paper, we present a novel technique for crafting adversarial perturbations using a random search optimization method guided by Grad-CAM [11]. By using Grad-CAM, we are able to identify critical positions that are likely to result in higher attack success rates. This enables us to focus on exploring a smaller set of positions during the random search, ultimately determining the best raindrop positions that ensure the highest effectiveness of our AdvRain attack. Our objective is to create perturbations that can be introduced into the visual path between the camera and the object, while keeping the object itself unaltered (As presented in Figure 3). By leveraging grad-cam, we can identify critical positions in the image that significantly influence the decision-making process of the deep learning model. This information guides our search optimization method, helping us generate adversarial perturbations at those influential positions. As a result, the perturbations effectively deceive the target classifier without directly modifying the object. The advantage of our approach lies in its ability to subtly manipulate the visual information captured by the camera. The crafted adversarial perturbations blend seamlessly into the scene, remaining inconspicuous to human observers. Simultaneously, they have a substantial impact on the model's decision, leading to misclassification of the object.

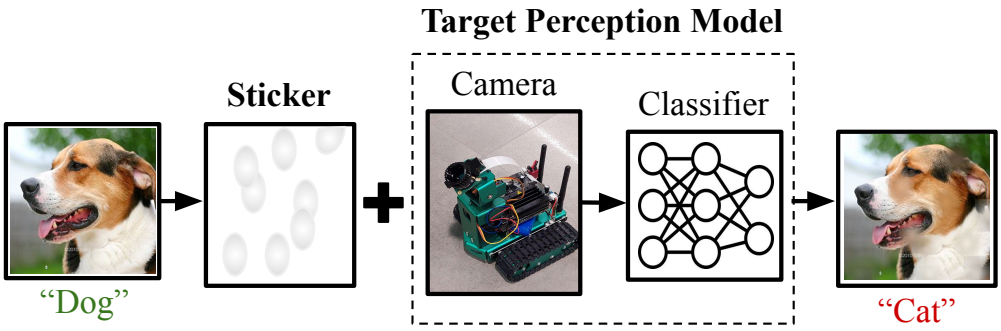

Figure 3. Attack threat model: The generated pattern is printed on a translucent sticker placed over the lens of the camera. Hence, any captured image will contain the adversarial dots resulting in an inconspicuous, natural-looking adversarial image that fools the target model and the human eye.

An overview of our novel contributions is shown in Figure 4.

In summary, the *contributions* of this work are:

- We propose a novel technique that utilizes a random search optimization method guided by grad-cam to craft adversarial perturbations. These perturbations are introduced into the visual path between the camera and the object without altering the appearance of the object itself.
- The adversarial perturbations are designed to resemble a natural phenomenon, specifically raindrops, resulting in an inconspicuous pattern. These patterns are printed on a translucent sticker and affixed to the camera lens, making them difficult to detect.
- The proposed adversarial sticker applies the same perturbation to all images belonging to the same class, making it a universal attack against the target class.
- Our experiments demonstrate the potency of the AdvRain attack, achieving a significant decrease of over 61% in accuracy for VGG-19 on ImageNet and 57% for Resnet34 on Caltech-101 compared to 37% and 40% (for the same structural similarity index (SSIM)) when using FakeWeather [9].
- We study the impact of blurring specific parts of the image, introducing low-frequency patterns, on model interpretability. This provides valuable insights into the behavior of the deep learning models under the proposed adversarial attack.

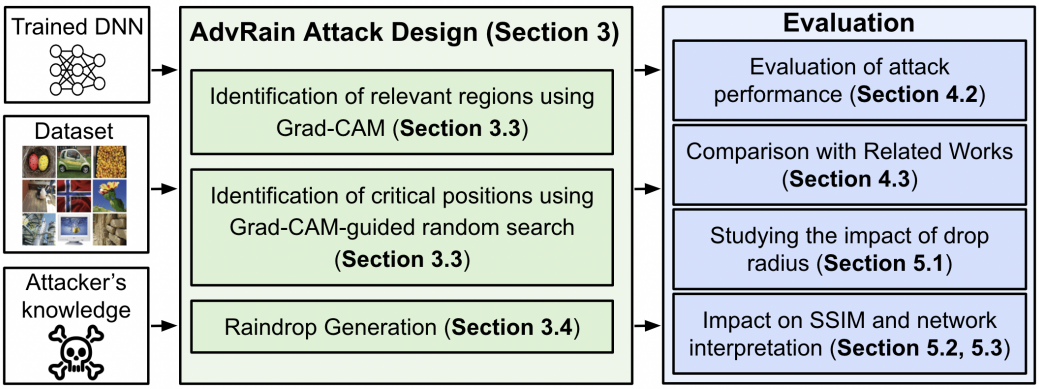

Figure 4. Overview of our Novel Contributions.

Paper Organization: The structure of the remaining article is as follows. Section 2 offers a comprehensive overview of related work, setting the context for our proposed approach. In Section 3, we present the methodology for generating the adversarial sticker, outlining the key components of our AdvRain attack. Next, in Section 4, we conduct a detailed evaluation of the proposed attack. We compare AdvRain against FakeWeather and Natural rain, examining the attack's potency and the visual similarity of the generated adversarial examples. Additionally, we examine the attack's potency and the visual similarity of the generated adversarial examples. In Section 5, we thoroughly discuss the findings and implications of our experiments. Additionally, we explore the impact of drop radius, assess the perturbation's effect on SSIM, and investigate its influence on network interpretation. Finally, in Section 6, we provide a concise summary and conclusion of our study, highlighting the significance of AdvRain as a stealthy and effective approach to adversarial attacks in camera-based vision systems.

2. Background & Related Work

2.1. Camera-Based Vision Systems

Environment perception has emerged as a crucial application, driving significant efforts in both the industry and research community. The focus on developing powerful deep learning (DL)-based solutions has been particularly evident in applications such as autonomous robots and intelligent transportation systems. Designing reliable recognition systems is among the primary challenges in achieving robust environment perception. In this context, automotive cameras play a pivotal role, along with associated perception models such as object detectors and image classifiers, forming the foundation for vision-based perception modules. These modules are instrumental in gathering essential information about the environment, aiding autonomous vehicles (AVs) in making critical decisions for safe driving.

The pursuit of dependable recognition systems represents a major hurdle in establishing high-performance environment perception. The accuracy and reliability of such systems are critical for the safe operation of AVs and ensuring their successful integration into real-world scenarios. However, it has been demonstrated that these recognition systems are susceptible to adversarial attacks, which can undermine their integrity and pose potential risks to AVs and their passengers. In light of these challenges, addressing the vulnerabilities of DL-based environment perception systems to adversarial attacks becomes paramount. Research and development efforts must focus on building robust defense mechanisms to fortify these systems against potential threats, enabling the safe and trustworthy deployment of AVs and intelligent transportation systems in the future.

Adversarial attacks can be categorized into digital and physical attacks.

2.2. Digital Adversarial Attacks

In scenarios involving digital attacks, the attacker has the flexibility to manipulate the input image of a victim deep neural network (DNN) at the pixel level. These attacks assume that the attacker has access to the DNN's input system, such as a camera or other means of providing input data. The concept of adversarial examples, where a small, imperceptible noise is injected to shift the model's prediction towards the wrong class, was first introduced by Szegedy et al. [12].

Over time, various algorithms for creating adversarial examples have been developed, leading to the advancement of digital attacks. Some notable digital attacks include Carlini and Wagner's attack (CW) [5], Fast Gradient Sign Method (FGSM) [3], Basic Iterative Method (BIM) [13], local search attacks [14], and HopSkipJump attack (HSJ) [15], among others. However, in a realistic threat model, we may assume that the attacker has control over the system's external environment or external objects, rather than having access to the system's internal sensors and data pipelines. This scenario represents physical attacks, where the attacker crafts adversarial perturbations that can be introduced into the physical world to deceive the DNN.

In the following section, we will explore some state-of-the-art physical attacks on image classification. These attacks are designed to exploit the vulnerabilities of camera-based vision systems and demonstrate the potential risks of adversarial manipulation in real-world scenarios. Understanding and mitigating such physical attacks are crucial for building robust and secure intelligent systems, particularly in safety-critical applications such as autonomous vehicles and surveillance systems.

2.3. Physical Adversarial Attacks

A physical attack involves adding perturbations in the physical space to deceive the target model. The process of crafting a physical perturbation typically involves two main steps. First, the adversary generates an adversarial perturbation in the digital space. Then, the goal is to reproduce this perturbation in the physical space, where it can be perceived by sensors such as cameras and radars, effectively fooling the target model.

Existing methods for adding adversarial perturbations in different locations can be categorized into four main groups: *Attack by Directly Modifying the Targeted Object*: In this approach, the attacker directly modifies the targeted object to introduce the adversarial perturbation. For example, adversarial clothing has been proposed, where clothing patterns are designed to confuse object detectors [16–18]. Hu et al. [16] leverage pretrained GAN models to generate realistic/naturalistic images that can be printed on t-shirts and are capable of hiding the person wearing them. Guesmi et al. [17] proposed replacing the GAN with a semantic constraint based on adding a similarity term to the loss function and, in doing so, directly manipulating the pixels of the image. This results in a higher flexibility to incorporate multiple transformations.

Attack by Modifying the Background: Adversarial patches represent a specific category of adversarial perturbations designed to manipulate localized regions within an image, aiming to mislead classification models. These attacks leverage the model's sensitivity to local alterations, intending to introduce subtle changes that have a substantial impact on the model's predictions. By exploiting the model's reliance on specific image features or patterns, adversaries can create patches that trick the model into misclassifying the image or perceiving it differently from its actual content. An example of a practical attack for real-world scenarios is AdvPatch [10]. This attack creates universal patches that can be applied anywhere. Additionally, the attack incorporates Expectation over Transformation (EOT) [19] to enhance the robustness of the adversarial patch. The AdvCam technique [8] presents an innovative approach to image perturbation, operating within the style space. This method combines principles from neural style transfer and adversarial attacks to craft adversarial perturbations that seamlessly blend into an image's visual style. For instance, AdvCam can introduce perturbations such as rust-like spots on a stop sign, making them appear natural and inconspicuous within their surroundings.

Modifying the Camera: This method involves modifying the camera itself to introduce the adversarial perturbation. One approach is to leverage the Rolling Shutter Effect, where the timing of capturing different parts of the image is manipulated to create perturbations [20,21].

Modifying the Medium Between the Camera and the Object: This category includes attacks that modify the medium between the camera and the object. For instance, light-based attacks use external light sources to create perturbations that are captured by the camera and mislead the target model [6,22]. The Object Physical Adversarial Device (OPAD) [22] employs structured lighting methods to alter the appearance of a targeted object. This attack system is composed of a cost-effective projector, a camera, and a computer, enabling the manipulation of real-world objects in a single shot. Zhong et al. [6] harness the natural phenomenon of shadows to create adversarial examples. This method is designed to be practical in both digital and physical contexts. Unlike traditional gradient-based optimization algorithms, it employs optimization strategies grounded in particle swarm optimization (PSO) [23]. The researchers conducted extensive assessments in both simulated and real-world scenarios, revealing the potential threat posed by shadows as a viable avenue for attacks. However, it is important to note that these techniques may experience reduced effectiveness under varying lighting conditions. The Adversarial Color Film (AdvCF) method, introduced by Zhang et al. [7], utilizes a color film positioned between the camera lens and the subject of interest to enable effective physical adversarial attacks. By adjusting the physical characteristics of the color film without altering the appearance of the target object, AdvCF aims to create adversarial perturbations that maintain their effectiveness in various lighting conditions, including both daytime and nighttime settings.

FakeWeather [9] attack aims to emulate the effects of various weather conditions, such as rain, snow, and hail, on camera lenses. This attack seeks to deceive computer vision systems, particularly those used in autonomous vehicles and other image-based applications, by adding perturbations to the captured images that mimic the distortions caused by adverse weather. In the FakeWeather attack, the adversary designs specific masks or patterns that simulate the visual artifacts produced by different weather conditions. These masks are then applied to the camera's images, introducing distortions that can mislead image recognition models. The goal is to make the images appear as if they were captured in inclement weather, potentially causing the models to make incorrect predictions or classifications. One limitation of the FakeWeather attack is that the generated noise or perturbations may have unrealistic and pixelated patterns, which could potentially be detected by more robust image recognition systems. Additionally, the attack's effectiveness may be limited to specific scenarios and image sizes, as it was initially tested on small images of 32×32 pixels from the CIFAR-10 dataset.

Additionally, researchers have explored techniques to create adversarial perturbations with natural styles to ensure stealthiness and legitimacy to human observers. Such approaches aim to make the perturbations appear as natural phenomena in the scene.

3. Proposed Approach

In this section, we outline our methodology for designing the adversarial camera sticker, a novel approach to creating inconspicuous adversarial perturbations for camera-based vision systems.

3.1. Threat Model for Physical Camera Sticker Attacks

Traditionally, the problem of generating an adversarial example is formulated as a constrained optimization (Equation (1)), given an original input image x and a target classification model $f(.)$:

$$\min_{\delta} \|\delta\|_p s.t. f(x+\delta) \neq f(x) \tag{1}$$

where the objective is to find a minimal inconspicuous universal perturbation, δ, such that when added to an arbitrary input from a target input domain D, it will cause the underlying DNN-based model $f(.)$ to misclassify. Note that one cannot find a closed form solution for this optimization problem since the DNN-based model $f(.)$ is a non-convex machine learning model, i.e., a deep neural network. Therefore, Equation (1) is formulated as follows to numerically solve the problem using empirical approximation techniques:

$$\arg\max_{\delta} \sum_{x \in \mathcal{D}} l(f(x+\delta), f(x)) \tag{2}$$

where l is the DNN-based model loss function and $\mathcal{D} \subset D$ is the attacker's classifier training dataset. To solve this problem, existing optimization techniques (e.g., Adam [24]) can be used. In each iteration of the training the optimizer updates the adversarial noise δ.

In contrast to attacks that operate at the pixel-level granularity of the images, our proposed threat model, the physical camera sticker attacks, faces a significant challenge due to the limited space of feasible perturbations that can be introduced. The optics of the camera impose constraints, resulting in the generation of only blurry dots as perturbations, lacking the high-frequency patterns typically found in traditional adversarial attacks. To address this challenge, we propose an innovative approach to design the adversarial perturbation by approximating the effect of placing small dots, resembling raindrops, on a sticker. When a small water drop is placed on the camera lens, it creates a translucent patch on the captured image. This patch represents the introduced low-frequency perturbations resulting from the optics of the camera lens. The adversarial example is crafted as follows:

$$x_{adv} = G(x, p, n, r) \tag{3}$$

where G is the raindrop generator, p stands for the drops positions, n for number of drops, r for the drop radius. Technically, considering a 2D image x with $x(i,j)$ denoting the pixel at the (i,j) location. Our aim is to find the best position candidate $p(i,j)$ for the n raindrops, in a way that the model wrongly classifies the generated adversarial example.

$$p(i,j)? \; s.t. \; f(G(x, p(i,j), n, r)) \neq f(x) \tag{4}$$

3.2. Overview of the Proposed Approach

Figure 5 provides an overview of our proposed attack. The main objective is to generate adversarial perturbations with patterns resembling the effect of natural weather events, particularly raindrops on the camera lens due to atmospheric conditions (i.e., Rain). To achieve this, we craft patterns that simulate the appearance of raindrops on the camera lens.

AdvRain utilizes a unique approach to create adversarial perturbations that mimic the effect of natural weather conditions, particularly rain, on camera lenses. This approach can be broken down into several key steps: The process begins with the generation of a pattern that resembles the appearance of raindrops on a camera lens. These raindrops are a form of low-frequency perturbations that blur portions of the captured image. The goal is to make these perturbations inconspicuous and blend them seamlessly with the visual characteristics of real raindrops.

To optimize the pattern of raindrops for maximum effectiveness, a random search guided by Grad-CAM (Gradient-weighted Class Activation Mapping) is employed. Grad-CAM helps identify important regions within the image that influence the model's decision. By starting the optimization process from positions highlighted by Grad-CAM, the search narrows down and explores a smaller set of potential perturbation patterns. Once a pattern is selected, it is applied to the input image as simulated raindrops. Each pixel within the area covered by a raindrop undergoes a transformation, specifically a Gaussian blur. This blurring effect mimics the distortion introduced by raindrops on a camera lens. In addition to the blur, AdvRain also incorporates a fish-eye effect to simulate the curvature that raindrops can introduce to the camera's view. The splashing of raindrops

is simulated using collision detection, ensuring that the raindrops do not overlap in a physically implausible manner.

To create a natural raindrop appearance, AdvRain employs a combination of a circle and an oval shape for each raindrop. The final shape represents the water droplet's surface, and the blur applied to this shape adds to the realism. The generated perturbation, resembling a collection of raindrops, is intended to be printed on a translucent sticker that can be affixed to the camera's lens. In essence, AdvRain leverages a combination of random search optimization, Grad-CAM guidance, and the transformation of carefully designed raindrop patterns to create physically inconspicuous adversarial perturbations. These perturbations, once applied to the camera lens, introduce low-frequency distortions that can deceive deep neural networks into misclassifying images.

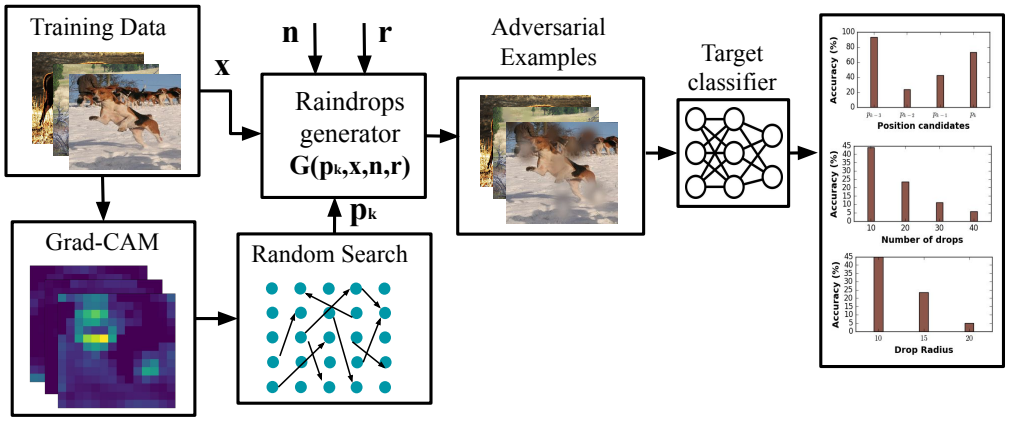

Figure 5. Overview of the proposed approach; **Training phase:** we employ a random search-based optimization method and leverage Grad-CAM to enhance the efficiency of the optimization process. By using Grad-CAM, we are able to identify critical positions that are likely to result in higher attack success rates. This enables us to focus on exploring a smaller set of positions during the random search, ultimately determining the best raindrop positions that ensure the highest effectiveness of our AdvRain attack. The raindrop generator takes as inputs the number of drops n, the radius of the drop r, the positions p_k of the n drop for the k iteration, and the input image x. The output of the generator is the simulated raindrops added to the input image. This image is later on fed to the classifier to monitor its classification accuracy.

3.3. Identify Critical Positions

The identification of critical positions is achieved using a Grad-CAM-guided Random Search Method, as outlined in Algorithm 1. The main objective of this optimization process is to select the best position candidate for introducing raindrop perturbations that result in minimal classification accuracy (i.e., maximum attack success rate).

Our proposed search method is presented in Algorithm 1: We start by initializing the number of iterations T and the number of candidate positions N. We then initialize the perturbation as an empty set. For each iteration t from 1 to T, we randomly generate N candidate positions within the image. For each candidate position, we apply the Gaussian blur and fish-eye effect to create the raindrop pattern. We simulate the effect of the raindrop perturbation on the image using the candidate position and use Grad-Cam to identify the most critical regions for the model's decision. Then, we select the candidate position that results in the highest attack success rate (i.e., the lowest classification accuracy). After that, we update the perturbation with the raindrop pattern from the selected candidate position. We then return the final perturbation, which represents the optimal raindrop pattern for introducing adversarial perturbations with minimal classification accuracy.

This grad-cam-guided random search-based optimization method efficiently explores different positions for introducing the raindrop perturbations, focusing on regions that significantly influence the model's decision. By selecting the position that maximizes the

attack success rate, we ensure that the adversarial camera sticker is effective in deceiving the target deep learning model while appearing inconspicuous in the captured images.

Algorithm 1 Grad-CAM-guided Random Search Method

```
Initialize the number of iterations T and the number of candidate positions N
Initialize the perturbation as an empty set.
Use Grad-Cam to identify the most critical regions for the model's decision.
for t = 1 : T do
    Randomly generate N candidate positions within the identified critical regions.
    for p = 1 : N do
        Apply the Gaussian blur and fish-eye effect to create the raindrop pattern.
        Simulate the effect of the raindrop perturbation on the image using the candidate position.
        Select the candidate position that results in the highest attack success rate.
        Update the perturbation with the raindrop pattern from the selected candidate position.
    end for
end for
```

3.4. Raindrop Generator

The drop generator applies a Gaussian blur transformation to each pixel belonging to the area covered by the raindrop. The transformation function is defined as follows: Let $I(i,j)$ be the input image, where (i,j) are the pixel coordinates, and $I'(i,j)$ be the resulting image after applying the Gaussian blur transformation for the raindrop.

The Gaussian blur transformation is given by

$$I'(i,j) = \frac{1}{\sum_{a,b} W(n,m)} \sum_{a,b} I(i+a, j+b)\dot{W}(a,b) \tag{5}$$

where $W(a,b)$ is the Gaussian kernel defined as

$$W(a,b) = \frac{1}{2\pi\sigma^2} \exp - \frac{a^2+b^2}{2\sigma^2} \tag{6}$$

here, a and b are the pixel offsets within the raindrop area, and σ is the standard deviation of the Gaussian kernel, controlling the blur intensity.

The application of the Gaussian blur transformation ensures that the raindrop pattern appears natural and seamless in the image, emulating the effect of raindrops on the camera lens due to atmospheric conditions. This approach contributes to the inconspicuousness of the adversarial perturbation, making it difficult to detect by human observers while effectively deceiving the target deep learning model.

To create the realistic effect of raindrop surfaces, our approach involves both Gaussian blur and a fish-eye effect. The process of simulating raindrops and their splashing is achieved through collision detection, where we check if the center of a raindrop overlaps with another raindrop. If there is an overlap, the raindrops are merged; otherwise, no action is taken. To craft the shape of raindrops, we utilize a combination of one circle and one oval (Figure 6a). By manipulating the size and orientation of these shapes, we can create different gaps and simulate the appearance of water droplets (Figure 6b). The final effect of the water droplet surface is achieved through the application of Gaussian blur (Figure 6c). The generated perturbation, representing the raindrop patterns, is then printed on a translucent sticker. This sticker is carefully affixed to the camera lens, allowing the perturbations to be introduced into the visual path without obstructing the image capture process.

The combination of the fish-eye effect, collision detection, and raindrop shape design, along with the Gaussian blur, results in realistic and inconspicuous raindrop patterns on

the camera lens. These patterns effectively mislead the target deep learning model while resembling a natural weather phenomenon. This approach demonstrates the effectiveness of our proposed AdvRain attack, highlighting its potential impact on camera-based vision systems and the significance of robust defense mechanisms to counter physical adversarial attacks in real-world applications. Figure 6 showcases the concept and various stages of raindrop pattern creation.

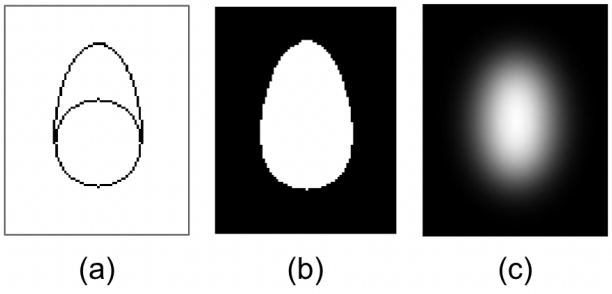

Figure 6. Raindrop generation process: (**a**) The raindrop shape is formed using one circle and one oval. (**b**) The final shape. (**c**) We then create the effect of the water droplet surface through adding the blur effect.

4. Experimental Results

4.1. Experimental Setup

In our study, we conducted experiments to evaluate the impact of different sizes of raindrops on the classification accuracy of VGG-19 [25] and Resnet34 [26], which serve as the victim classifiers. The input image resolution for both models is set to 224 × 224 pixels.

For our evaluations, we utilized images from two well-known datasets: ImageNet and Caltech-101 [27]. ImageNet [28] is a large visual dataset containing over 14 million annotated images, organized into more than 20,000 categories. On the other hand, Caltech-101 consists of images depicting objects from 101 distinct classes, with approximately 9000 images in total. Each class contains a varying number of images, ranging from 40 to 800, and the images have variable sizes with typical edge lengths of 200–300 pixels. Throughout our experiments, we established the baseline classification accuracy of the models by evaluating their performance when fed with clean, unaltered images. By simulating different sizes of raindrops and studying their effects on model classification accuracy, we gained valuable insights into the robustness of the models under adversarial perturbations. The comparison to the baseline classification accuracy allowed us to quantitatively assess the effectiveness of our AdvRain attack and its ability to deceive the models while introducing minimal visual changes to the images. The experimental setup is summarized in Figure 7.

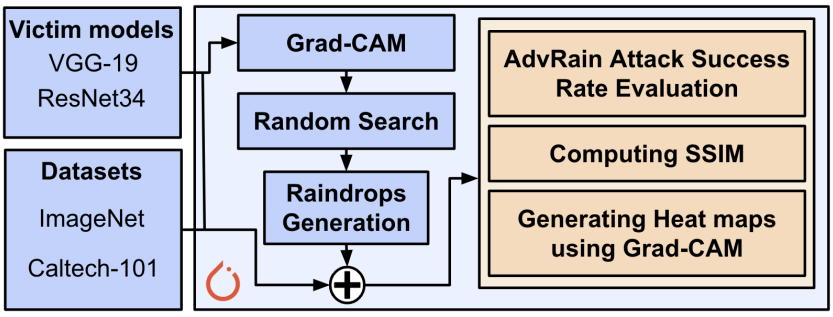

Figure 7. Experimental setup and tool-flow for conducting our experiments.

4.2. Evaluation of Attack Performance

In our evaluation, we adopted classification accuracy as the primary metric to assess the effectiveness of our AdvRain attack. To demonstrate the impact of the attack, we first

generated the adversarial perturbation for 10 different classes and then measured the model accuracy for various numbers of raindrops.

The results of our experiments revealed that our attack successfully reduced the classification accuracy of the models. Specifically, with just 10 raindrops applied, we observed a significant decrease in accuracy of approximately 30% for the ImageNet dataset and 20% for the Caltech-101 dataset. As the number of raindrops increased to 20, the drop in accuracy became more pronounced, reaching more than 40% for ImageNet and over 30% for Caltech-101 (as depicted in Table 1).

Table 1. Impact of drop radius on model accuracy for VGG-19 trained for ImageNet and Resnet34 trained on Caltech-101: The bigger the radius of the drop the higher drop in classification accuracy we achieve.

Number of Drops	VGG-19	Resnet34
0	100%	100%
10	72%	81%
20	59%	69%
30	46%	50%
40	39%	43%

These findings indicate the potency of the AdvRain attack in misclassifying objects of different classes while maintaining inconspicuous patterns resembling natural weather events. The considerable reduction in model accuracy with just a small number of raindrops highlights the susceptibility of camera-based vision systems to physical adversarial attacks, underscoring the need for robust defense strategies to counter such threats in real-world applications.

In Table 2, we present the per-class classification accuracy for eight different classes: Walker hound, Cock, Snake, Spider, Fish, Parrot, American flamingo, and Bison. Among these classes, the "Walker hound" class exhibited the highest drop in accuracy when subjected to our AdvRain attack. This can be attributed to the fact that the ImageNet dataset includes a total of 120 categories of dog breeds, many of which have close features and similarities. Consequently, the presence of raindrop perturbations in the images of Walker hounds, which share visual characteristics with other dog breeds, led to a more substantial decrease in classification accuracy.

Table 2. Average Model Accuracy per class for different number of raindrops (Drop radius = 10) for VGG-19 on ImageNet.

Number of Drops	Walker Hound	Cock	Snake	Spider	Fish	Parrot	Flamingo	Bison
10	43%	70%	79%	68%	66%	86%	78%	68%
20	22%	56%	63%	59%	58%	82%	69%	57%
30	12%	48%	49%	51%	35%	68%	60%	33%
40	4%	40%	37%	43%	29%	60%	52%	31%

However, for classes with more distinguishable and unique features, such as the "Parrot" class and the "American flamingo", we observed a relatively smaller drop in accuracy. Fooling these objects proves to be more challenging due to their distinctive visual attributes, which made it harder for the raindrop perturbations to significantly mislead the models.

These per-class results provide valuable insights into the impact of our AdvRain attack on different object categories. The varying degrees of accuracy drop across classes underscore the importance of considering the visual characteristics and complexity of objects when assessing the effectiveness of adversarial attacks. Such knowledge is crucial for understanding the limitations and strengths of the attack and can guide the development

of targeted defense strategies for enhancing the robustness of camera-based vision systems in the face of physical adversarial perturbations.

4.3. Comparison with State-of-the-Art: AdvRain vs. FakeWeather

In this section, we conduct a comparison between our AdvRain attack and the FakeWeather attack proposed in [9]. The FakeWeather attack aims to simulate the effects of rain, snow, and hail on camera lenses by creating three masks that mimic the appearance of these weather conditions. However, a key limitation of the FakeWeather attack is the use of unrealistic and pixelated patterns for generating noise. Additionally, the added perturbations cover a significant portion of the image (Figure 8), potentially making them more conspicuous and easily detectable by human observers.

Furthermore, the effectiveness of the FakeWeather attack was evaluated solely on small images with a size of 32×32 pixels from the Cifar-10 dataset. In contrast, our proposed AdvRain attack generates more realistic perturbations that closely emulate the effect of raindrops on camera lenses. The raindrop patterns crafted by our attack are inconspicuous and blend seamlessly into the image, making them challenging to detect visually.

Moreover, we extend the evaluation of AdvRain to larger images with a size of 224×224 pixels from the ImageNet and Caltech-101 datasets. This broader evaluation showcases the versatility and robustness of our attack on high-resolution images, making it more suitable for real-world scenarios and camera-based vision systems, such as those used in autonomous vehicles and intelligent robots.

As illustrated in Table 3, our propose attack outperforms FakeWeather when used to attack models trained on ImageNet and Caltech-101. For instance, AdvRain achieves an attack success rate of 65%; however, for the same SSIM, FakeWeather achieves only 37%.

In summary, our AdvRain attack outperforms the FakeWeather attack by producing more realistic and inconspicuous raindrop perturbations. The improved emulation of the rain effect and the demonstrated effectiveness on larger images strengthen the applicability and impact of our proposed attack in challenging the integrity of deep learning models in camera-based vision systems. This comparison emphasizes the significance of developing attacks that are not only powerful but also realistic and unobtrusive in real-world applications.

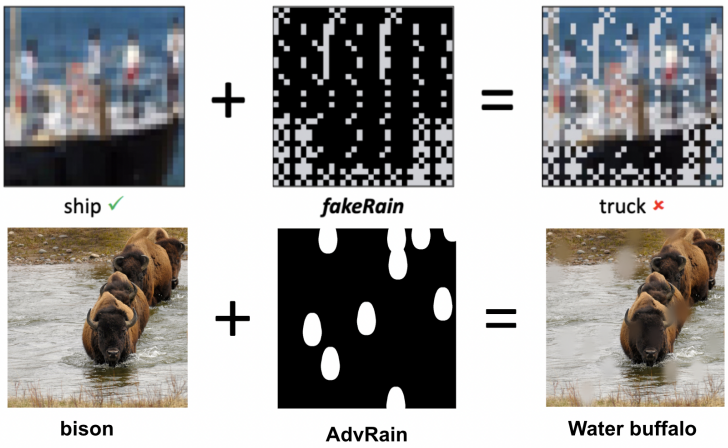

Figure 8. AdvRain compared to FakeWeather attack. FakeWeather attack tries to emulate the rain effect by designing a mask that fakes the effect of such weather conditions on the camera lenses by changing the pixel values. However, the generated mask resulted in unrealistic and pixelated patterns. In contrast, AdvRain is based on generating more realistic raindrops simulation with a shape closer to that of a real raindrop.

Table 3. Attack Success Rate: AdvRain vs. FakeWeather [9].

Method	ImageNet	Caltech-101
FakeWeather [9]	37%	40%
AdvRain (ours)	65%	62%

4.4. AdvRain vs. Natural Rain

In our evaluation, we conducted a comparison between our AdvRain attack and natural rain. To simulate natural rain, we randomly placed raindrops in the images, and this approach resulted in varying degrees of accuracy degradation for the victim model. As depicted in Figure 9, different positions of the randomly placed raindrops led to different accuracy drops.

The comparison highlights the effectiveness and superiority of our AdvRain attack over natural rain in achieving significant classification accuracy drops. For the left combination of totally random raindrops, the accuracy drop was only 3%, indicating that the random placement of raindrops had limited impact on fooling the victim model. However, when we carefully selected the positions of raindrops using our AdvRain attack, the accuracy drop reached a significant value of 60%.

These results demonstrate that the adversarial perturbations crafted by our AdvRain attack are much more potent in deceiving the deep learning model compared to the impact of natural rain. The strategic placement of raindrops in AdvRain leads to a higher level of misclassification, indicating that our attack successfully leverages the characteristics of raindrop patterns to create more effective adversarial perturbations.

This comparison emphasizes the unique advantage of AdvRain in generating inconspicuous yet powerful adversarial perturbations that effectively mislead the target model, surpassing the impact of random raindrop placements associated with natural rain. The ability to achieve substantial accuracy drops with carefully selected raindrop positions highlights the potential real-world implications of AdvRain, particularly in scenarios involving camera-based vision systems such as those used in autonomous vehicles and intelligent robots.

Figure 9. Impact of AdvRain on classification accuracy compared to natural rain. *Left:* Natural rain (Random positioning of the raindrops) resulted in a 3% drop. *Middle:* The combination of non-optimal positions resulted in a drop of 23%. *Right:* the carefully selected combination (AdvRain's output) resulted in a 60% drop.

5. Discussion

5.1. Evaluation on More DNN Models

AdvRain attack has been tested on other convolutional neural network (CNN) models, specifically Resnet-50 and Inception v4. As shown in Table 4, when AdvRain is applied to Resnet-50, it successfully causes misclassification in the target model in approximately 59% of the cases. In other words, when images perturbed by AdvRain are fed into Resnet-50,

the model's accuracy drops by 59% compared to its accuracy on clean, unperturbed images. This demonstrates the effectiveness of the AdvRain attack on Resnet-50. Additionaly, when AdvRain is applied to the Inception v4 model, it achieves a success rate of approximately 57%. This indicates that the attack causes a 57% drop in accuracy when applied to Inception v4. This highlights the performance of the AdvRain attack on different CNN models, specifically Resnet-50 and Inception v4. It shows the effectiveness of the attack, which represents the extent to which AdvRain can deceive or mislead these models when applied to images.

Table 4. Performance of AdvRain on More CNN models.

Model	VGG-19	Resnet-50	Inception v4
Accuracy	37%	41%	43%

5.2. Impact of Drop Radius

In our evaluation, we systematically investigated the impact of changing the radius of the raindrop (i.e., the blurred area) on the effectiveness of our AdvRain attack. We tested three different models trained on ImageNet (i.e., VGG-19, Resnet-50, Inception v4) and three others trained on Caltech-101 (i.e., Resnet-34, Resnet-50, and Inception v4). We observed that increasing the radius led to a substantial increase in the attack success rate, further compromising the accuracy of the victim model. For instance, when we increased the radius from 10 to 15 pixels for the ImageNet dataset, the accuracy of the victim model dropped an additional 25% (as depicted in Tables 5 and 6). This finding demonstrates the significance of the raindrop size in determining the strength of the adversarial perturbations.

Table 5. Impact of drop radius on model accuracy for ImageNet: The bigger the radius of the drop, the higher drop in classification accuracy we achieve.

Number of Drops	0	10		20	
Radius	-	10	15	10	15
VGG-19	100%	76%	57%	59%	37%

The larger the radius of the raindrop, the more extensive the area covered by the perturbation, making it more impactful in fooling the deep learning model. The increased attack success rate with larger raindrop sizes indicates that our AdvRain attack is particularly effective in situations where the raindrop pattern covers a significant portion of the image, emulating the effect of real-world raindrops on the camera lens.

Table 6. Impact of drop radius on model accuracy for Caltech-101: The bigger the radius of the drop the higher drop in classification accuracy we achieve.

Number of Drops	0	10		20	
Radius	-	10	15	10	15
Resnet-34	100%	79%	70%	69%	55%

5.3. Impact on SSIM

In this section, we focus on evaluating the impact of the adversarial perturbation on the captured images using the structural similarity index measure (SSIM) [29]. Given that our AdvRain attack is designed to introduce raindrop perturbations by blurring specific regions of the image, we observe a relatively small impact on SSIM when compared to traditional adversarial attacks (Table 7). For instance, when using 10 raindrops, the SSIM of the adversarial example compared to the clean image is measured at 0.89. This indicates that

the introduction of raindrop perturbations has a limited effect on the structural similarity between the adversarial and clean images.

The relatively high SSIM values suggest that our AdvRain attack achieves its goal of creating inconspicuous perturbations that closely resemble natural weather phenomena. Despite causing a significant drop in classification accuracy, the raindrop perturbations retain a high degree of structural similarity to the original image, making them difficult for human observers to detect visually. This aspect sets our AdvRain attack apart from traditional adversarial attacks that often introduce noticeable and disruptive patterns in the image, resulting in lower SSIM values. The inconspicuous nature of our perturbations contributes to the stealthiness and effectiveness of AdvRain in deceiving camera-based vision systems without significantly altering the visual appearance of the captured images.

Table 7. SSIM of adversarial examples compared to the original images.

Number of Drops	10	20	30	40
SSIM	0.89	0.78	0.73	0.69

5.4. Impact on Network Interpretation

Adversarial patches have demonstrated their efficacy in causing misclassification, but their usage has also raised concerns about their detectability. Standard network interpretation methods, such as Grad-CAM, can highlight the presence of adversarial patches, effectively disclosing the identity of the adversary [11]. Grad-CAM, being one of the most well-known network interpretation algorithms, has proven to outperform other state-of-the-art interpretation methods in various scenarios.

To evaluate the Grad-CAM visualization results for traditional adversarial patches versus our AdvRain attack, we utilized an ImageNet pre-trained VGG-19 classifier. The evaluation involved comparing the effects of adding low frequency patterns (traditional adversarial patch) versus high frequency patterns (AdvRain) on the model's interpretation.

Unlike patch-based attacks that shift the model's focus from the object to the location of the patch, potentially making them detectable, our AdvRain attack exhibits a different behavior. It causes the model to overlook some crucial features that are essential for the model to make accurate decisions (as illustrated in Figure 10). By strategically introducing raindrop perturbations, we divert the model's attention away from vital features, making it more susceptible to misclassification.

This behavior reinforces the stealthiness of our AdvRain attack since it does not draw attention to a specific patch or region, unlike traditional adversarial patches. The inconspicuous nature of the raindrop perturbations, combined with their impact on the model's interpretation, highlights the effectiveness of AdvRain in deceiving camera-based vision systems without raising suspicions about the presence of an adversarial attack.

The Grad-CAM visualization results provide further evidence of the unique advantages of our AdvRain attack, emphasizing its potential as a powerful and inconspicuous technique for crafting adversarial perturbations that elude standard detection methods and undermine the reliability of deep learning models in real-world applications.

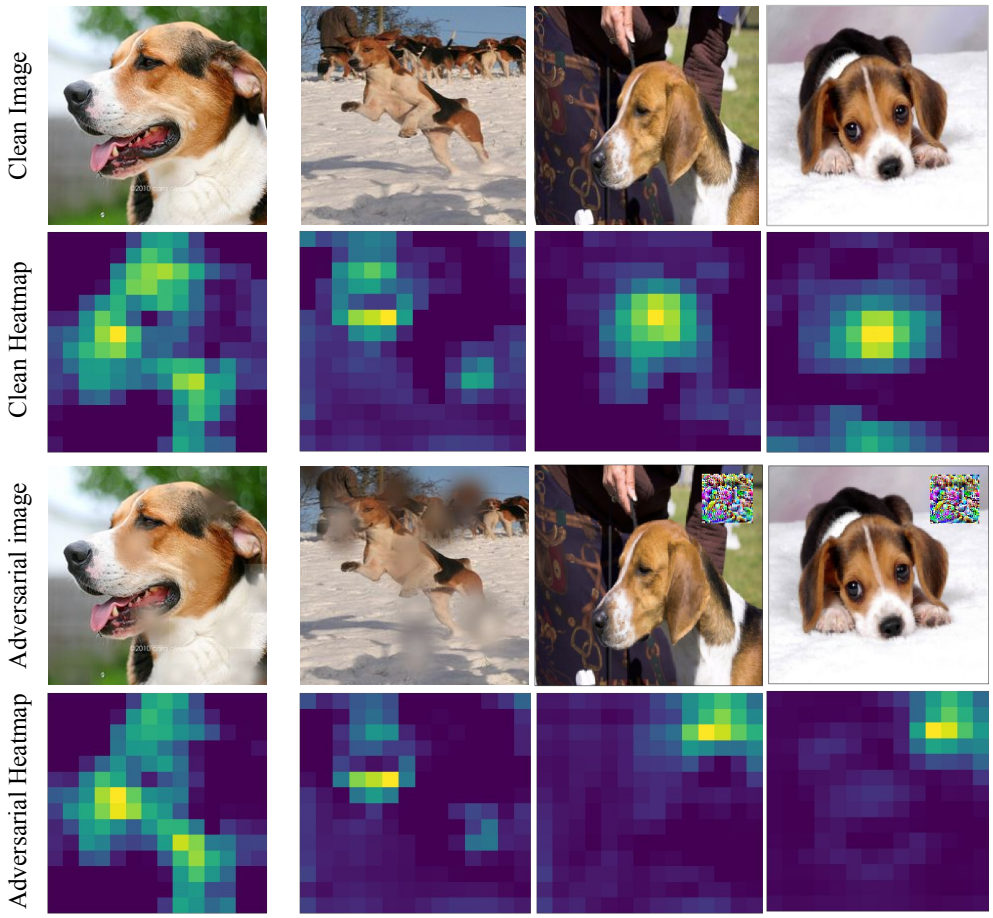

Figure 10. Comparing the Grad-CAM visualization results for adversarial patch vs. AdvRain.

5.5. Possible Defenses

One potential defense strategy against AdvRain involves training a denoising/deraining model. This approach aims to enhance the resilience of the DNN against adversarial attacks of this nature. The denoiser operates as a preprocessing stage, effectively eliminating the adversarial noise from the input data before it reaches the target model. This denoising model is specifically designed to mitigate the impact of AdvRain perturbations.

The denoising/deraining model can be designed to learn how to cancel or neutralize the influence of AdvRain perturbations on the model's output. This can be achieved by defining a loss function that guides the training of the denoising model. Here is how this process can work: The loss function can be defined to measure the dissimilarity between the predictions made by the victim model when given a clean image and when given a denoised (derained) image. The goal is to minimize this dissimilarity, effectively forcing the denoising model to remove AdvRain perturbations in a way that ensures the victim model's output remains consistent with that of a clean image. The training data will consist of pairs of images: Clean Images (i.e., images that have not been subjected to AdvRain attacks) and AdvRain-Infected Images (i.e., images that have been attacked by AdvRain to introduce perturbations).

The denoising model is trained on these pairs of images to minimize the defined loss function. During training, the model learns to identify and understand the patterns associated with AdvRain perturbations. It also learns how to modify the AdvRain-infected image to cancel out these perturbations, effectively generating a denoised version that closely resembles the clean image. After training, when the denoising model receives an AdvRain-infected image as input, it applies the learned transformations to cancel out the AdvRain perturbations. The derained image, which is free of AdvRain noise, is then passed

to the victim model for classification or other tasks. Because the denoising model has effectively removed the AdvRain influence, the victim model's output remains consistent with what it would produce when given a clean image. This approach enhances the robustness of the system against AdvRain attacks. The denoising model acts as a filter that "cleans" AdvRain-infected images before they reach the victim model, ensuring that the model's predictions are not adversely affected by the perturbations.

Another defense strategy can be to generate adversarial samples considering different rain models, and incorporate these samples during the adversarial training process for a given DNN. By training the DNN on this augmented dataset that encompasses a variety of AdvRain-induced adversarial samples, the model can become more robust and resilient to AdvRain attacks across a wider range scenarios.

5.6. AdvRain on Videos

One notable feature of AdvRain is its ability to conduct class-wide attacks. This means that the same adversarial perturbation, represented by the raindrop pattern on the sticker attached to the camera of a system (e.g., cars), can be universally applied to different images of objects within the same class. This universal approach simplifies the attack process and efficiently deceives the target model across a broad range of objects within the targeted class. Regarding the application of AdvRain in videos, its effectiveness depends on several factors, with context stability being a key consideration. We believe that AdvRain can maintain its effectiveness in video scenarios where the context remains relatively stable. In such cases, the attack can continue to deceive the model effectively in video scenario. This phenomenon is commonly observed in video scenes characterized by consistent visual elements.

5.7. Limitations

AdvRain is an effective and stealthy approach to adversarial attacks, but like any technique, it may have some potential limitations, as discussed below:

- **Device-Specific**: The success of AdvRain can depend on the specific camera and lens characteristics of the target device. Variations in camera types, lens coatings, and sensor resolutions can affect the effectiveness of the attack. This means that the same AdvRain sticker may not work with same efficiency on all devices.
- **Sticker Placement**: The success of the attack depends on the precise placement of the AdvRain sticker on the camera lens. If the sticker is misaligned or partially obstructed, it may not create the desired perturbations, reducing the attack's effectiveness.
- **Environmental Conditions**: The effectiveness of AdvRain can be influenced by environmental conditions such as lighting, weather, and visibility. Raindrop patterns may be more or less convincing under different conditions, potentially limiting the attack's reliability.

6. Conclusions

In this paper, we present a novel approach for crafting realistic physical adversarial camera stickers to deceive image classifiers. Our proposed attack, known as AdvRain, emulates the effect of natural weather conditions, specifically raindrops, when placed on the camera lens. Our methodology revolves around utilizing random search-based optimization methods to identify the optimal raindrop positions. Through extensive evaluations, we demonstrate the effectiveness of AdvRain. With just 20 raindrops, we achieve a significant drop in average classification accuracy, exceeding 45% and 40% for VGG19 on ImageNet and Resnet34 on Caltech-101, respectively. Our results showcase the potency of AdvRain as a stealthy and powerful approach to adversarial attacks in camera-based vision systems.

Author Contributions: Conceptualization, A.G., M.A.H. and M.S.; methodology, A.G., M.A.H. and M.S.; software, A.G.; validation, A.G.; formal analysis, A.G.; investigation, A.G.; resources, A.G.; data curation, A.G.; writing—original draft preparation, A.G., M.A.H. and M.S.; writing—review and editing, M.S.; visualization, A.G.; supervision, M.S.; project administration, M.S.; funding acquisition, M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in parts by the NYUAD Center for Cyber Security (CCS), funded by Tamkeen under the NYUAD Research Institute Award G1104, Center for Interacting Urban Networks (CITIES), funded by Tamkeen under the NYUAD Research Institute Award CG001, and Center for Artificial Intelligence and Robotics (CAIR), funded by Tamkeen under the NYUAD Research Institute Award CG010.

Data Availability Statement: No new data were created.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Al-Qizwini, M.; Barjasteh, I.; Al-Qassab, H.; Radha, H. Deep learning algorithm for autonomous driving using GoogLeNet. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, 11–14 June 2017; pp. 89–96.
2. Tramèr, F.; Zhang, F.; Juels, A.; Reiter, M.K.; Ristenpart, T. Stealing Machine Learning Models via Prediction APIs. In Proceedings of the 25th USENIX Security Symposium (USENIX Security 16), Austin, TX, USA, 10–12 August 2016; pp. 601–618.
3. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and Harnessing Adversarial Examples. *arXiv* **2015**, arXiv:1412.6572.
4. Kurakin, A.; Goodfellow, I.J.; Bengio, S. Adversarial examples in the physical world. *arXiv* **2016**, arXiv:1607.02533.
5. Carlini, N.; Wagner, D.A. Towards Evaluating the Robustness of Neural Networks. *arXiv* **2016**, arXiv:1608.04644.
6. Zhong, Y.; Liu, X.; Zhai, D.; Jiang, J.; Ji, X. Shadows can be Dangerous: Stealthy and Effective Physical-world Adversarial Attack by Natural Phenomenon. *arXiv* **2022**, arXiv:2209.02430.
7. Hu, C.; Shi, W. Adversarial Color Film: Effective Physical-World Attack to DNNs. *arXiv* **2022**, arXiv:2209.02430.
8. Duan, R.; Ma, X.; Wang, Y.; Bailey, J.; Qin, A.K.; Yang, Y. Adversarial Camouflage: Hiding Physical-World Attacks with Natural Styles. *arXiv* **2020**, arXiv:2003.08757.
9. Marchisio, A.; Caramia, G.; Martina, M.; Shafique, M. fakeWeather: Adversarial Attacks for Deep Neural Networks Emulating Weather Conditions on the Camera Lens of Autonomous Systems. *arXiv* **2022**, arXiv:2205.13807.
10. Brown, T.; Mane, D.; Roy, A.; Abadi, M.; Gilmer, J. Adversarial Patch. *arXiv* **2017**, arXiv:1712.09665.
11. Subramanya, A.; Pillai, V.; Pirsiavash, H. Towards Hiding Adversarial Examples from Network Interpretation. *arXiv* **2018**, arXiv:1812.02843.
12. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.J.; Fergus, R. Intriguing properties of neural networks. In Proceedings of the 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, 14–16 April 2014.
13. Brendel, W.; Rauber, J.; Bethge, M. Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models. *arXiv* **2017**, arXiv:1712.04248.
14. Narodytska, N.; Kasiviswanathan, S.P. Simple Black-Box Adversarial Perturbations for Deep Networks. *arXiv* **2016**, arXiv:1612.06299.
15. Chen, J.; Jordan, M.I. Boundary Attack++: Query-Efficient Decision-Based Adversarial Attack. *arXiv* **2019**, arXiv:1904.02144.
16. Hu, Y.C.T.; Chen, J.C.; Kung, B.H.; Hua, K.L.; Tan, D.S. Naturalistic Physical Adversarial Patch for Object Detectors. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 7828–7837. [CrossRef]
17. Guesmi, A.; Bilasco, I.M.; Shafique, M.; Alouani, I. AdvART: Adversarial Art for Camouflaged Object Detection Attacks. *arXiv* **2023**, arXiv:2303.01734.
18. Guesmi, A.; Ding, R.; Hanif, M.A.; Alouani, I.; Shafique, M. DAP: A Dynamic Adversarial Patch for Evading Person Detectors. *arXiv* **2023**, arXiv:2305.11618.
19. Athalye, A.; Engstrom, L.; Ilyas, A.; Kwok, K. Synthesizing robust adversarial examples. *PMLR* **2018**, *80*, 284–293.
20. Sayles, A.; Hooda, A.; Gupta, M.; Chatterjee, R.; Fernandes, E. Invisible Perturbations: Physical Adversarial Examples Exploiting the Rolling Shutter Effect. *arXiv* **2020**, arXiv:2011.13375.
21. Kim, K.; Kim, J.; Song, S.; Choi, J.H.; Joo, C.; Lee, J.S. Light Lies: Optical Adversarial Attack. *arXiv* **2021**, arXiv:2106.09908.
22. Gnanasambandam, A.; Sherman, A.M.; Chan, S.H. Optical Adversarial Attack. *arXiv* **2021**, arXiv:2108.06247.
23. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the Proceedings of ICNN'95—International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948. [CrossRef]
24. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
25. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
26. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *arXiv* **2015**, arXiv:1512.03385.
27. Li, F.F.; Andreeto, M.; Ranzato, M.; Perona, P. *Caltech 101*; CaltechDATA; California Institute of Technology: Pasadena, CA, USA, 2022. [CrossRef]

28. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
29. Wang, Z.; Bovik, A.; Sheikh, H.; Simoncelli, E. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef]