Texture- and Shape-based Adversarial Attacks for Vehicle Detection in Synthetic Overhead Imagery

Mikael Yeghiazaryan[1] Sai Abhishek Siddhartha Namburu[1] Emily Kim[1] Stanislav Panev[1]
Celso de Melo[2] Brent Lance[2] Fernando De la Torre[1] Jessica K. Hodgins[1]
[1]Carnegie Mellon University [2]Army Research Lab
myeghiaz@andrew.cmu.edu, siddharthanamburu@gmail.com, {ekim2, spanev}@andrew.cmu.edu
{celso.m.demelo.civ, brent.j.lance.civ}@army.mil
{ftorre, jkh}@cs.cmu.edu

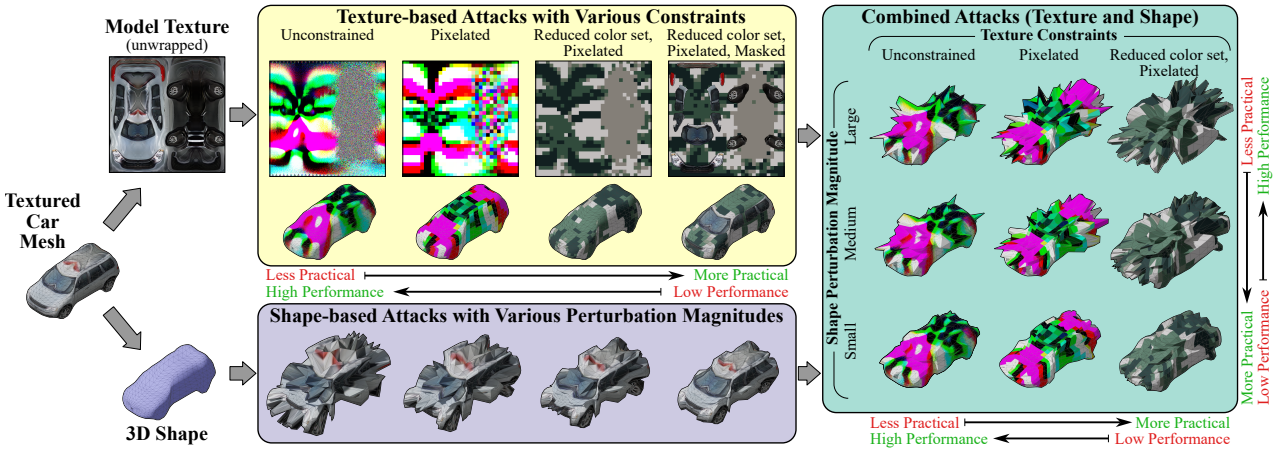

Figure 1. We enforced various constraints on common adversarial attacks to improve their implementation in the real world. Our pipeline perturbs objects' texture, shape, or both. The latter demonstrates superior efficacy in deceiving object detectors.

Abstract

Detecting vehicles in aerial images can be very challenging due to complex backgrounds, small resolution, shadows, and occlusions. Despite the effectiveness of SOTA detectors such as YOLO, they remain vulnerable to adversarial attacks (AAs), compromising their reliability. Traditional AA strategies often overlook the practical constraints of physical implementation, focusing solely on attack performance. Our work addresses this issue by proposing practical implementation constraints for AA in texture and/or shape. These constraints include pixelation, masking, limiting the color palette of the textures, and constraining the shape modifications. We evaluated the proposed constraints through extensive experiments using three widely used object detector architectures, and compared them to previous works. The results demonstrate the effectiveness of our solutions and reveal a trade-off between practicality and performance. Additionally, we introduce a labeled dataset of overhead images featuring vehicles of various categories. We will make the code/dataset public upon paper acceptance.

1. Introduction

Robust object detection in aerial and satellite images is vital for automating critical tasks like traffic management, urban planning, and disaster response. State-of-the-art (SOTA) detectors such as YOLO and RetinaNet, which rely on deep neural networks (DNNs), have become foundational in this domain. However, recent studies such as Szegedy *et al*. [74] have revealed that DNNs can be susceptible to adversarial examples. Given the importance of these applications, understanding this vulnerability is crucial, especially in object detection in Remote Sensing Imagery (RSI). Furthermore, there are scenarios where utilizing AAs to impede vehicle detection by computer vision systems in overhead images could offer strategic advan-

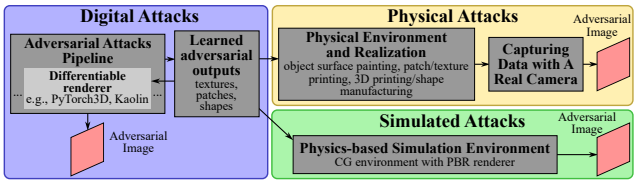

Figure 2. Illustration of the three types of adversarial attacks.

tages, such as military camouflage.

Our primary objective is to investigate the resilience of object detectors against adversarial vehicles in RSI scenarios under realistic constraints. Traditional AA strategies often neglect the constraints of physical implementation, focusing solely on task performance. For example, adversarial texture patterns typically resemble those depicted in Figure 1 (unconstrained texture-only attack). However, it is essential to consider how these patterns could be realistically applied, such as attaching a printed vinyl wrap to a vehicle. In this study, we propose a set of constraints within AAs to ensure the attack is feasible for implementation on vehicles, effectively camouflaging them.

We define a *practical adversarial mesh* as a mesh that is perturbed from its original state (by modifying texture and/or shape) such that replicating the same set of modifications in real life would not require specialized equipment or significant resources, or at least would facilitate the process of implementing these modifications. We outline the following aspects that define practicality: installation cost, installation difficulty, and difficulty of operation. Our results indicate that while constrained attacks are less successful than traditional AAs [38, 72, 73, 75], they provide much better practicality of implementation. Shape-only attacks exhibit lower effectiveness than unconstrained texture attacks. Nevertheless, combining constrained texture with practical shape modifications enhances performance, reaching levels comparable to unconstrained texture attacks, see Figure 1.

Our work contributes to the field in several ways. (1) We introduce constrained AAs for shape and/or texture, designed to create practical camouflages capable of deceiving object detectors in RSI. These constraints facilitate more straightforward implementation compared to unconstrained camouflages. (2) We thoroughly examine how practicality and adversarial performance relate, finding that they have an inverse relationship. (3) We provide a new object detection dataset comprising labeled real overhead images featuring eight distinct vehicle classes covering a broad spectrum of commonly encountered categories. (4) We developed a tool for generating synthetic overhead images, contributing to creating synthetic datasets.

2. Related Work

Over the past ten years, adversarial attacks (AAs) have become an increasingly important research topic in computer vision. Szegedy *et al*. [74] addressed the problem of neuron explainability but sparked a new branch of research that targeted the stability and robustness of a range of computer vision models [19, 20, 30–32, 36, 41, 43, 46, 48, 52, 61, 62, 65, 67]. Recent research has proposed various methods for generating adversarial examples [13, 16, 33, 49, 55, 56]. This direction of research evolved into the class of *digital AAs* [23, 45, 50, 53, 58, 64, 76, 78], which according to [3] are defined as attacks where the attacker is capable of directly modifying image pixels to fool the victim model (see Figure 2). These attacks are usually imperceptible. A very different type of AAs is the class of *physical AAs* [9, 11, 15, 24, 27–29, 68, 69], which according to [3] are defined as attacks where the attacker has no direct access to the images supplied to the model, but threatens the model performance by physically manipulating the objects expected to appear in the target images. Some works attempt to perform imperceptible physical attacks, but the majority of such attacks are perceptible. We also highlight that there are works that lie at the intersection of digital and physical attacks, which we call *simulated AAs* [7, 8, 25, 35, 42, 51, 66, 75], because the majority of these works perform perceptible attacks but test them in simulated realistic environments, avoiding any real tests. Because we utilize synthetically generated data for testing, our work belongs to this class of AAs. Additionally, we utilize a realistic physics-based rendering to test our results, which also separates us from prior works. Moreover, considerable effort has been invested in crafting adversarial 3D geometry, as demonstrated in [44, 79, 82, 83]. Most of the works that generate adversarial shapes concentrate on point clouds. This focus is a direct consequence of extensive research in autonomous driving, where LIDARs play a crucial role in localization and mapping. Some works also conduct AAs in a coupled LIDAR+RGB setting [12]. However, it differs from our task, as we assume only RGB data in RSI. Unlike our work, most of the studies highlighted above completely avoid constraining the adversarial search space or use "light" constraints expressed as regularization terms in the loss functions.

Recent years have seen increased attention by the research community to *remote sensing imagery* [2, 6, 26, 40, 70, 81, 87, 89]. These images represent a top-down view and typically have resolutions ranging from several centimeters to several meters or even tens of meters per pixel. The exponential growth of this type of data necessitates robust automated solutions, which directly motivate adversarial attacks — a tool for identifying vulnerabilities in deep learning models. Although the research community has mostly paid attention to ground-level AAs due to their rel-

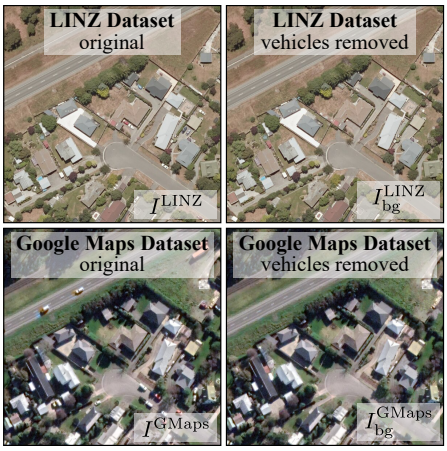

Figure 3. Comparison between LINZ and GMaps images reveals differences in resolution, color distribution, time of day, year, camera type, altitude (GMaps uses satellite, LINZ uses aerial), and post-processing. Both datasets had vehicles' silhouettes inpainted to blend with their surroundings.

evance to the safety-critical field of autonomous driving, there is a list of notable works in the field of RSI AAs [1, 4, 14, 18, 22, 38, 54, 80, 84, 85, 88]. However, most of these attacks are physically challenging to implement. To our knowledge, the only study implementing an aerial view adversarial attack is [22]. They employed diverse configurations of adversarial patches, such as placing them on the vehicle or its surroundings, to target vehicle detectors in images. Our approaches differ in the types of constraints applied to the adversarial patches (camouflages in our case) and their stringency.

3. Overview

Our work focuses on merging RSI with practical AAs on object detection models. We aim to develop realistic adversarial camouflages and vehicle modifications feasible for implementation, employing a novel set of constraints that mimic real-world limitations. While our constraints are not exhaustive, they are tailored to address critical limitations. Ideally, one would target all available model architectures, such as CNNs and ViTs [21]. However, for the sake of demonstration, we limit the scope of architectures. We detail our approach in the following sections, covering datasets (Section 4) and technical methodology (Section 5).

4. Datasets

This section describes the details of the datasets used for training, attacking, and evaluation. Section 4.1 describes the datasets of real images, and Section 4.2 describes the types of synthetic data we use, and their generation processes. Please see the supplementary material for examples.

4.1. Real Datasets

This paper introduces two novel real overhead image datasets named LINZ and GMaps. While both datasets cover the same geographical areas, they exhibit distinct visual differences, as seen in Figure 3. Further examples are shown in the supplementary material.

4.1.1 LINZ Dataset

To build the *labeled LINZ dataset*, we retrieved LINZ Data Service aerial orthoimages for the Selwyn District Council area [47], a high-resolution dataset of New Zealand's urban and suburban regions. We then manually labeled vehicle centers similar to [63]. Vehicle classes are: *Van RV*, *Truck*, *Bus*, *Small Trailer*, *Large Trailer*, *Specialized*, *Small Vehicle*, and *Unknown*. Most labels belong to the *Small Vehicle* class, encompassing common cars. The *Specialized* class includes harvesters and tractors, while *Unknown* encompasses unclassified vehicles, *e.g.* partially disassembled.

The original images downloaded from the LINZ Data Service website are too large for use in computer vision tasks. Thus, we sample them into smaller images of size $384\,\text{px} \times 384\,\text{px}$ resulting in $172\,595$ real images and use this image size throughout the paper. All datasets described in our work have the same geospatial resolution of $12.5\,\text{cm/px}$. Although we will make the entire labeled dataset public, we utilize only a specific subset of labels in our study. We choose the most common class, *Small Vehicles*, for all experiments, removing labels associated with other classes but preserving all images. This limitation gives greater clarity to our results' analysis. We denote this dataset with $\mathcal{I}^{\text{LINZ}}$.

We produced a second version of the dataset without vehicles (Figure 3) to serve as ground plane textures for the Blender synthetic dataset described below (Section 4.2). It is denoted as $\mathcal{I}^{\text{LINZ}}_{\text{bg}}$ and is obtained by utilizing *Inpaint Anything* by Yu *et al.* [86] along with already labeled ground-truth locations to automatically remove vehicles from $\mathcal{I}^{\text{LINZ}}$.

4.1.2 GMaps Dataset

The purpose of the GMaps dataset is to provide background images $\mathcal{I}^{\text{GMaps}}_{\text{bg}}$ for the PyTorch3D [60] data generation pipeline (see Section 4.2.1). To achieve this, we first extracted Google Maps (GMaps) [34] satellite images $\mathcal{I}^{\text{GMaps}}$ with matching geographical coordinates to the aerial LINZ dataset using QGIS [59], establishing a direct LINZ-GMaps correspondence. This process yielded the set of images $\mathcal{I}^{\text{GMaps}}$. We manually removed real vehicles from the original images using Adobe Photoshop [5]. Thus, we produced the set of background GMaps images $\mathcal{I}^{\text{GMaps}}_{\text{bg}}$. Later,

we used this set of images as background images in the synthetic data generation process described in Section 4.2.1.

4.2. Synthetic Data

We introduce two synthetic dataset generation pipelines that make use of PyTorch3D (PT3D) [60] and Blender [17]. The PT3D-based pipeline produces clean (non-adversarial) training data for the detectors and adversarial data on the fly during the attacks utilizing the differentiability of the PT3D renderer. On the other hand, the Blender-based pipeline generates realistic testing adversarial data to simulate real conditions, as the Cycles rendering engine [10] integrated into Blender produces physically plausible images.

4.2.1 PT3D Data Generation

The *PT3D data* is synthetically generated using PyTorch3D [60] as the renderer. We also need a set of 3D vehicle assets to generate realistic synthetic overhead images with vehicles. Instead of relying on manually created ones, we utilize GAN-based 3D mesh generators. Considering the low resolution typical of RSI, the meshes need not be overly detailed. In addition, identical UV maps for all meshes facilitate the application of unified adversarial texture maps. We use the Textured 3D GAN (T3GAN) model by Pavllo *et al*. [57]. However, due to the needs of our pipeline, we modify their code to enable semantic segmentation map sampling. In addition, we further train their mesh generator model to produce better semantic maps for the meshes. This procedure yields a set of meshes denoted as $\mathcal{M}$.

Given the GMaps backgrounds dataset $\mathcal{I}_{\text{bg}}^{\text{GMaps}}$ and the set of vehicle meshes $\mathcal{M}$, we use PyTorch3D's differentiable renderer (DR) to produce the images $I = \text{DR}\left(I_{\text{bg}}^{\text{GMaps}}, M\right)$, where I is the rendered image, $I_{\text{bg}}^{\text{GMaps}}$ is a background image sampled from $\mathcal{I}_{\text{bg}}^{\text{GMaps}}$ and M is a set of meshes randomly sampled from $\mathcal{M}$, which utilizes two main components: texture map T and shape information S (vertices, edges, and polygons). After rendering I, we apply several post-processing steps, such as blurring the vehicles and applying anti-aliasing, to improve the realism of the produced images. Additionally, we restrict M to contain only original or adversarial meshes. In the former case, the resulting image I is an "original" image, while it is an "adversarial" image in the latter case.

Following the above equation, we can generate synthetic datasets and their ground-truth annotations. We use a dataset of original PT3D images to train synthetic models. We also use the same pipeline for producing images on the fly during the adversarial attacks by simply replacing the set of shapes S and texture maps T with the optimized adversarial shapes S_{adv} and/or texture map T_{adv}.

4.2.2 Blender Data Generation

We use the *Blender data* to test the adversarial meshes in a more realistic scenario. We generate a test set for each adversarial texture or geometry using realistic Blender Cycles [10] physics-based renderer. Unlike PyTorch3D, Blender renders much more realistic data. However, because it is non-differentiable, we do not use it in the adversarial optimization process. To produce a synthetic overhead image with Blender, we use an image sampled from $\mathcal{I}_{\text{bg}}^{\text{LINZ}}$ and a set of meshes M, which can consist of either original meshes or meshes utilizing adversarial texture and/or shape.

5. Method

This section outlines the methodology for crafting adversarial vehicles. We provide an overview of the general pipeline employed for generating adversarial vehicles in Section 5.1. We delve into the specific techniques used to create adversarial textures and shapes in Sections 5.2 and 5.3, respectively. Section 5.4 elaborates on the procedure for simultaneously optimizing the texture and shape descriptors of a mesh.

5.1. General Pipeline

During each cycle of the attack, we use PT3D to generate a batch of adversarial images $I_b = \{I_1, \ldots, I_{N_b}\}$ of size N_b, such that $I_k = \text{DR}\left(I_{\text{bg},k}^{\text{GMaps}}, M_k\right), \forall\, k \in [1, \ldots, N_b]$, where I_k is the k-th image in I_b, $I_{\text{bg},k}^{\text{GMaps}}$ is the k-th image from the batch of background images $I_{\text{bg}}^{\text{GMaps}}$ sampled from the GMaps dataset $\mathcal{I}_{\text{bg}}^{\text{GMaps}}$, M_k is a randomly sampled mesh from $\mathcal{M}$ with its corresponding shape and texture components S_k and T_k respectively, and DR is a differentiable renderer as described in Section 4.2.1. Depending on the optimized entity, either the most recent S_{adv} or T_{adv} replaces the corresponding counterpart in M_k at the beginning of each iteration. See an overview of our pipeline in Figure 4.

During the attack, we ensure that each image contains only one vehicle. We adopt this limitation intentionally to prevent the production of meshes that may rely on multiple camouflaged vehicles being in close proximity to each other. Instead, we aim to produce adversarial meshes that are independently effective.

Let F_i be the objective function used to train a detector model D_i. We supply the batch of images I_b to D_i, producing a set of predictions $y_{\text{pred}} = D_i\left(I_b\right)$. We then minimize a weighted loss function for an ensemble of synthetic models:

$$\mathcal{L} = \sum_i \lambda_i \mathbb{E}\left[F_i\left(D_i\left(I_b\left(I_{\text{bg}}^{\text{GMaps}}, M\right)\right), y_{\text{gt}}\right)\right], \quad (1)$$

$$M_{\text{adv}}^{\star} = \arg\min_M \mathcal{L}\left(M\right), \quad (2)$$

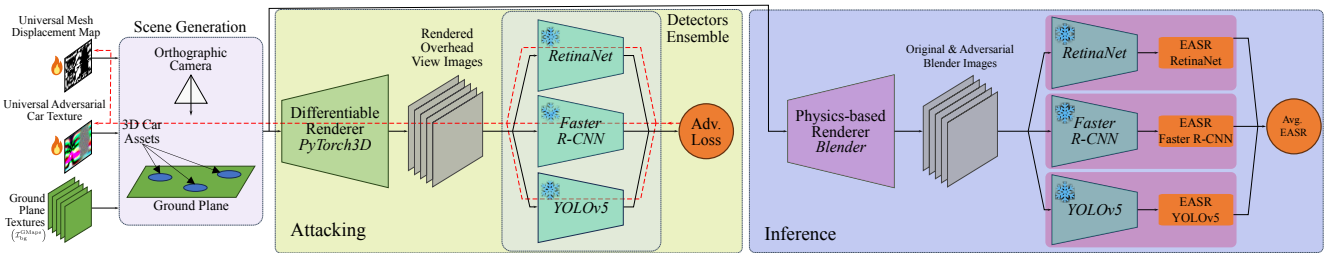

Figure 4. Our pipeline for adversarial attacks on an ensemble of object detectors. The back-propagation path is illustrated by the red dashed line. During inference, we make predictions using only one model. We then average the resulting EASRs for all models to analyze the results. Inference with PT3D data follows the same diagram, but instead of using Blender, we use PT3D to render the inference images.

where y_{gt} is the set of ground-truth locations of objects in the given image and is manually replaced with $y_{\text{gt}} = \emptyset$. D_i represents the i-th model from an ensemble of detection models trained on synthetic data. Depending on the optimized entity, either the S or the T component of M is optimizable. For each experiment, model coefficients λ_i are selected such that the initial loss values F_i for each model fall within the same order of magnitude. We recognize that there are numerous ways to select coefficients. However, we focus only on this one in our study.

5.2. Texture-based Attacks

In texture-based attacks, only one universal texture map is optimized, *i.e.* an adversarial texture map applied to all the meshes. While unconstrained adversarial textures (abbreviated as "U") achieve excellent performance in fooling vehicle detectors, they are far from being practical for implementation. We introduce constraints imposed on textures that allow the production of practical camouflages by reflecting limitations associated with the real world implementation. These constraints include the following: *Spatial Resolution*, *Spatial Restriction*, and *Color Restriction*.

Spatial Resolution. Applying iridescent patterns to irregular shapes such as vehicle surfaces is often challenging [4, 22, 37, 72, 73, 75, 77]. Hence, we adopt the first constraint expressed as texture pixelization (abbreviated as "Pix") with block sizes of $16\,\text{px} \times 16\,\text{px}$, corresponding to approximately $15\,\text{cm} \times 15\,\text{cm}$ on the rooftop of a vehicle. Given that RSI typically does not offer resolutions beyond $15\,\text{cm/px}$, this constraint does not noticeably impact the vehicle's image appearance, significantly streamlining the implementation process. We implement this constraint by storing a latent representation of the adversarial texture as a $32 \times 32 \times 3$ tensor. Upon texture generation request, we upscale this tensor to $512 \times 512 \times 3$ using the nearest-neighbor interpolation, resulting in a pixelated output.

Spatial Restriction. Another notable limitation is the necessity for vehicle camouflage not to hinder road visibility. Therefore, specific vehicle areas like glass must remain free of camouflage, which we address by segmenting the meshes used for the attacks into semantic parts. We then apply masks (abbreviated as "Ma") to restrict the area where the adversarial texture is applied. Hence, the second constraint is spatial restriction. It balances reduced performance from a smaller camouflage coverage area with an enhanced operational experience. To implement this constraint, we use an adversarial texture map T_{adv}, an original texture map T_{or} of a vehicle, and a corresponding binary segmentation mask T_{mask}. Using these three entities, we produce the segmented adversarial texture map as $T_{\text{segmented}} = T_{\text{or}} \cdot (1 - T_{\text{mask}}) + T_{\text{adv}} \cdot T_{\text{mask}}$.

Color Restriction. While the previous two limitations deal with placing textures on the vehicle surface, the final and strictest constraint remains color limitations. We leverage this constraint in two ways: 1) fixing the color count and enabling the attacker to learn optimal colors (abbreviated as "Lc") and their placement in the adversarial texture map, or 2) fixing the colors (abbreviated as "Fc") and letting the attacker learn their placement in the texture map. This constraint stands apart from softer constraints used in previous works, such as *non-printability score*, as it imposes a strict limitation on the number of colors in the output. It is crucial to recognize that while certain constraints mentioned earlier have been investigated in prior studies [22, 72, 73, 75], the concept of a "color restriction" remains largely unexplored in the literature. Furthermore, previous research efforts have not adequately addressed the simultaneous integration of the abovementioned constraints.

Finding an adversarial texture map with constrained colors involves determining the color of each pixel. At each attack iteration, we predict a probability distribution $p(c)$ for each pixel over a set of colors c. We then amplify the most probable color while damping the others using a softmax function $s(\cdot)$ twice: $p_A(c) = s(s(p(c)))$. The pixel color is set to $\mathbb{E}\left[p_A(c)\right]$, which is close to the mode color due to amplification. After optimization, each pixel is assigned $\arg\max_{c_i} p(c_i)$, resulting in a camouflage that meets the color constraint and is similar to the optimized camouflage which uses $\mathbb{E}\left[\cdot\right]$ instead of $\arg\max$. For more details on our constraints, please refer to the supplementary material.

Figure 5. Examples of adversarial images are shown, with the top row displaying the original images and the bottom row showing the corresponding adversarial (T-PixFcMa) images produced using Blender.

5.3. Shape-based Attacks

In shape-based attacks, only a universal shape perturbation is optimized. We parameterize the geometry deformations using a single-channel 2D displacement map generated from the common UV map of all the meshes. In contrast to the general definition of displacement maps where each pixel represents a deformation in normal, tangent, and bi-tangent directions, we apply deformations to mesh vertices in the direction defined by the vector joining the geometrical center of the mesh to the corresponding vertex. For each mesh, we compute the geometrical center as the mean 3D coordinate. When generating deformed meshes, the displacement map is interpolated to generate deformations at all vertices of a mesh. Moreover, the deformations are restricted to be positive to prevent altering the original vehicle mesh, and instead attaching shapes to it.

Similar to the texture-based attacks, we introduce constraints to enforce practicality. These constraints include *Symmetry* and *Perturbation Magnitude* (PM). The first constraint ensures that the resulting shapes have sideways balanced mass distribution which is important for manufacturing and vehicle operation. We implement this constraint by introducing two symmetry axes around which the displacement map is mirrored. The second constraint reduces the volume of the deformations. The allowed magnitude of perturbation for each mesh is defined as a fraction of the width of each car, where this fraction is defined by the hyper-parameter PM $\in [0, 1]$. We optimize for it by running multiple shape-based attacks and then choosing the most optimal one as described in Section 7, allowing us to balance between practicality and performance. For more details on the implementation of these constraints, see the supplementary material.

Table 1. Comparing the practicality of the attacks explored in our study to previous works. We do not distinguish between sequential and parallel combined attacks as they only impact the process, not the final result's form. The first symbol reflects the texture-related practicality score, the second number reflects the shape-related practicality score.

	Camouflage	PC	DI	DO	Score	Notes
Other works	Du *et al.* (ON) [22]	$+0$	$+0$	$+0$	$+3$	Small AA area
	Du *et al.* (OFF) [22]	00	$+0$	-0	0	Limited mobility
	EVD4UAV [71]	$+0$	$+0$	$+0$	$+3$	Small AA area
	FCA [75]	-0	-0	$+0$	-1	Special equipment
	ACTIVE [73]	-0	-0	$+0$	-1	Special equipment
	DTA [72]	-0	-0	$+0$	-1	Special equipment
Our	T-U	-0	-0	-0	-3	Special equipment
	T-Ma	-0	-0	$+0$	-1	Special equipment
	T-PixMa	-0	$+0$	$+0$	$+1$	Stickers/paint squares
	T-PixFcMa	$+0$	$+0$	$+0$	$+3$	Lim. color stickers
	S-O	$0-$	$0-$	$0-$	-3	Shape modification
	C-Fc	$--$	$--$	$--$	-6	Spec. eq./shape mod.
	C-PixFc	$+-$	$+-$	$--$	-2	Shape modification

5.4. Combined Attacks

We also conduct combined adversarial attacks, which result in vehicles with adversarial texture and shape. We split them into two branches: *sequential* and *parallel* depending on how the attacks are performed.

In *sequential combined adversarial attacks*, we borrow an adversarial texture map obtained from a texture-based attack, use it instead of vehicle texture, and perform a shape-based attack to optimize the displacement map described in Section 5.3. Thus, both mesh properties are attacked sequentially. The difference between this and the shape-based attacks is the initial texture map: instead of the original textures, an adversarial texture is utilized. In addition, similar to shape-based attacks, we run a series of experiments to establish the optimal PM for each sequential attack.

In *parallel combined adversarial attacks*, both texture and shape are optimized. We alternate between optimizing the adversarial texture map and the displacement map: optimizing one entity for a fixed number of steps n_{pll}, then switching to the other entity for another n_{pll} steps. We continue this process until the loss converges.

6. Practicality and Comparisons

Our primary focus is not to enhance the performance of AAs but to explore the impact of realistic constraints on them. Given the high EASR of unconstrained AAs, there is limited room for improvement. We assess our work against three qualitative criteria: *production cost* (PC), *difficulty of installation* (DI), and *difficulty of operation* (DO). Each criterion is rated as good ($+$), insignificant (0), or bad ($-$), with practical camouflages earning more good scores. Please refer to the supplementary material for a detailed definition of these criteria.

Based on the outlined criteria and as depicted in Ta-

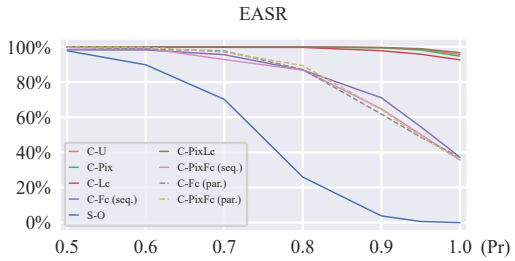

Figure 6. EASR vs Practicality (Pr) curves for the shape-based and combined attacks on PT3D.

ble 1, the most practical camouflage is the texture-based T-PixFcMa, despite having one of the lowest EASR (Section 7). The only works that have comparable practicality scores are Du *et al*. [22] (ON) and EVD4UAV [71]. However, the areas occupied by their patches are very small for being effective in aerial images of the resolution considered in our work. Please refer to the supplementary material for a more in-depth description of the score assignment. Results from Tables 1 to 2 illustrate the trade-off between practicality and performance. While some previous studies excel in AAs for vehicle detectors, they lack practicality. We conclude that performance and practicality are inversely related when optimization for performance is the goal. For example, for randomly generated camouflages (Table 2), the reduction in performance may not be justifiable because no optimization is carried out.

We recognize that this analysis could be carried out more rigorously and incorporate numerical comparisons from real data. For example, DO could be evaluated through extensive user studies. However, due to limited resources, we leave such analysis beyond the scope of the paper and rely on the assumptions made above.

7. Experiments and Results

7.1. Evaluation metrics

To evaluate the effectiveness of the adversarial meshes, we rendered two matched image datasets, D_{or} and D_{adv}, from each experiment. For D_{or}, we composed and rendered 3D scenes with the original car meshes. For D_{adv}, we apply adversarial modifications to the meshes of the same scenes. Thus, each image from D_{or} has an identical version (the same background, lighting, camera parameters, and car locations and orientations) with adversarial cars. We compute the percentage of vehicles **d**etected in D_{or} but **m**issed in D_{adv}. $V_{d,m}$ represents such vehicles, and $V_{d,d}$ denotes vehicles detected in both D_{or} and D_{adv}. This computation yields the *Attack Success Rate* $\text{ASR} = \frac{|V_{d,m}|}{|V_{d,d} \cup V_{d,m}|}$, where $|\cdot|$ is the cardinality operator. In our task, avoiding introducing new detections $V_{m,d}$ after applying the adversarial entity is also important. Thus, we modify ASR to account for this:

$$\text{EASR} = \frac{|V_{d,m}| - |V_{m,d}|}{|V_{d,d} \cup V_{d,m}|} = \text{ASR} - \text{ER}, \qquad (3)$$

where EASR is the *Effective Attack success Rate* and ER is the *erroneous rate*, *i.e*. fraction of true-positive detections that emerged after introducing the adversarial entity.

7.2. Detection Models

We used three model architectures for vehicle center detection in RSI, including RetinaNet [46], Faster R-CNN [62], and YOLOv5 [39]. We chose these models to represent YOLO-family detectors (YOLOv5), one-stage detectors (RetinaNet), and two-stage detectors (Faster R-CNN). Each architecture was trained on both real and synthetic datasets, resulting in models labeled as "real" and "synthetic" (abbreviated as "synt"). On the real test set, synthetic models achieve around 50%–63% average precision, while real models score above 80%. Detailed performance figures are available in the supplementary material. As expected, models perform better within their training domains. We attacked ensembles of synthetic models and used only one synthetic model for inference.

7.3. Texture-based Attacks

We alter a vehicle's texture in *texture-based* attacks, aiming to hide it from detectors. We initialize the adversarial textures randomly. Given the set of constraints described in Section 5.2, we conclude that there are twelve possible distinct adversarial texture settings (Table 2-upper half). In each setting, we attack the same ensemble of three synthetic models (RetinaNet, Faster R-CNN, YOLOv5) to obtain the complete set of adversarial textures. We also generate four random texture maps to compare the effectiveness of the adversarial textures to those of a random guess. See each texture setting, denoted as T-*, and the corresponding results on PT3D test data in Table 2. We also report the attack results using randomly generated textures, denoted as R-*. In Fc experiments, five colors were selected using K-means clustering on background image pixels $I_{\text{bg}}^{\text{GMaps}}$. In Lc experiments, the model learned the placement of five colors.

Recognizing the distribution gap between the real and the PT3D datasets, we repeat the evaluation experiments using Blender as described in Section 4.2.2. Using a computer graphics tool like Blender allows us to conduct tests with the camouflages in a more realistic simulated environment. See the results of tests on Blender data in Table 2. We conclude that constraints, as expected, reduce the performance but gain practicality (see Section 6). See example images in Figure 5. Please refer to the supplementary material for more examples. Additionally, we emphasize that the practicality-performance trade-off reveals itself even when

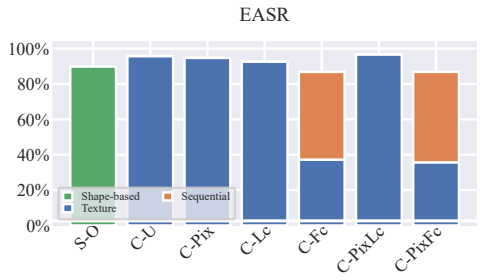

Figure 7. Gain in EASR due to geometric modifications.

Table 2. The figures show mean values from evaluations of individual synthetic models on PT3D and Blender data. "T", "R", "S" and "C" represent the texture, random texture, shape, and combined attacks. Note that Lc and Fc are mutually exclusive by definition. PM* and Pr* represent the optimal perturbation magnitude and practicality of the attacks involving shape modifications.

Attack	Constraints				PM*	Pr*	PT3D EASR	Blender EASR
	Pix	Lc	Fc	Ma				
T-U					—	—	95.77%	70.02%
T-Ma				✓	—	—	75.76%	44.43%
T-Pix	✓				—	—	94.75%	63.83%
T-PixMa	✓			✓	—	—	73.39%	41.49%
T-Lc		✓			—	—	92.63%	74.65%
T-Fc			✓		—	—	37.22%	55.96%
T-LcMa		✓		✓	—	—	71.93%	52.51%
T-FcMa			✓	✓	—	—	12.58%	26.43%
T-PixLc	✓	✓			—	—	96.73%	68.85%
T-PixFc	✓		✓		—	—	35.68%	55.87%
T-PixLcMa	✓	✓		✓	—	—	68.39%	42.15%
T-PixFcMa	✓		✓	✓	—	—	12.70%	44.64%
R-U					—	—	0.88%	16.31%
R-Pix	✓				—	—	3.86%	18.95%
R-Fc			✓		—	—	3.30%	18.95%
R-PixFc	✓		✓		—	—	3.16%	20.24%
S-O	—	—	—	—	0.4	0.6	89.82%	78.86%
C-U					0.0	1.0	96.51%	—
C-Pix	✓				0.0	1.0	95.02%	—
C-Lc		✓			0.0	1.0	92.68%	—
C-Fc (seq.)			✓		0.2	0.8	86.80%	70.37%
C-Fc (par.)			✓		0.2	0.8	87.11%	68.07%
C-PixLc	✓	✓			0.0	1.0	96.40%	—
C-PixFc (seq.)	✓		✓		0.2	0.8	86.83%	75.76%
C-PixFc (par.)	✓		✓		0.2	0.8	89.34%	77.86%

testing on a domain different from the optimized one, such as Blender data.

We examine color distribution in adversarial texture maps and find that lacking color constraints leads to saturated colors at RGB cube corners. This issue persists in prior works too. See the supplementary material for details.

7.4. Shape-based Attacks

In *shape-based attacks*, we alter the geometry of the vehicle sacrificing practicality for improved adversarial performance. We link practicality, denoted as Pr, to the perturbation magnitude PM as $\text{Pr} = 1 - \text{PM}$, where $\text{PM} \in [0, 1]$. In this context, $\text{Pr} \in [0, 1]$, with $\text{Pr} = 0$ indicating extreme mesh perturbation and $\text{Pr} = 1$ indicating no perturbation for a more realistic scenario.

Our goal is to minimize PM (maximize Pr) in shape-based attacks while maximizing EASR performance. We assess multiple attacks on synthetic models across a range of Pr values. Refer to the curve in Figure 6 under "Original" for details (utilizing original vehicle textures). Given an EASR-vs-Pr curve, we compute a practicality metric P1 as the harmonic mean of the EASR and Pr values for each point on the curve, *i.e.* $\text{P1} = 2\frac{\text{EASR}\cdot\text{Pr}}{\text{EASR}+\text{Pr}}$. We then pick the attack with the highest P1 as the optimal one. See the results, denoted as S-O, in Table 2 and Figure 7. The results suggest that when no adversarial texture is utilized along with a deformed vehicle geometry, the deformation must be large to achieve good performance, which is expensive to produce and difficult to install and operate, rendering it impractical.

7.5. Combined Attacks

In the *combined attacks*, we aim to boost the adversarial performance by attacking both mesh entities. We discard the adversarial textures that use masking because a modified mesh is hard to segment into semantically meaningful parts. Thus, this leaves us with six adversarial texture maps for mesh-based attacks, each with its texture setting.

Sequential Attacks. We conduct six sequential attacks, each using one of the six adversarial texture maps from a non-masked setup. We follow the methodology in Section 7.4 to determine the optimal PM. The results, labeled C-*, in Table 2 and Figure 7, reveal significant insights. While the improvement over texture attacks without the fixed colors constraint (T-U, T-Pix, T-Lc, T-PixLc) is unjustified due to practicality loss, the fixed colors constraint (T-Fc and T-PixFc) justifies sacrificing some practicality for better performance. Compared to the significant practicality loss in shape-based attacks, the sequential blend of adversarial texture and shape is more efficient than shape-only attacks. We evaluate the resulting adversarial vehicles on Blender-generated data where the optimal perturbation magnitude is not 0.

Parallel Attacks. We conduct parallel attacks using only two adversarial texture maps — Fc and PixFc — to reduce the PM even further than the sequential attacks. The results in Table 2 suggest no significant gain in switching to parallel attacks.

8. Conclusion

In summary, our study introduces a methodology for crafting efficient camouflage strategies for concealing vehicles in RSI. While our findings could potentially be misused, we believe the research community must be informed and address the critical vulnerabilities in state-of-the-art models highlighted in our work. We demonstrate an inverse relationship between practicality and performance, which

generalizes across domains. Unconstrained adversarial textures are potent in attacking vehicle detectors, whereas practical constrained textures are less effective but easier to implement. Practical shape-only attacks are slightly less effective than texture attacks. Combining texture and shape modifications can enhance performance to levels comparable to unconstrained textures. Notably, sequential and parallel execution of shape and texture attacks yield similar adversarial performance. Furthermore, we present an innovative labeled dataset comprising real-world aerial images and introduce two tools designed to generate synthetic aerial images: one using a differentiable renderer and one using a physics-based renderer.

References

[1] A. Van Etten. The Weaknesses of Adversarial Camouflage in Overhead Imagery. In *2022 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, pages 1–7, Los Alamitos, CA, USA, oct 2022. IEEE Computer Society. 3

[2] Lydia Abady, János Horváth, Benedetta Tondi, Edward J Delp, and Mauro Barni. Manipulation and generation of synthetic satellite images using deep learning models. *Journal of Applied Remote Sensing*, 16(4):046504–046504, 2022. 2

[3] Syed M. Kazam Abbas Kazmi, Nayyer Aafaq, Mansoor Ahmad Khan, Ammar Saleem, and Zahid Ali. Adversarial Attacks on Aerial Imagery : The State-of-the-Art and Perspective. In *2023 3rd International Conference on Artificial Intelligence (ICAI)*, pages 95–102, 2023. 2

[4] Ajaya Adhikari, Richard J. M. den Hollander, Ioannis Tolios, Michael van Bekkum, Anneloes Bal, Stijn Hendriks, Maarten Kruithof, Dennis Gross, Nils Jansen, Guillermo A. Pérez, Kit Buurman, and Stephan Raaijmakers. Adversarial Patch Camouflage against Aerial Detection. *CoRR*, abs/2008.13671, 2020. 3, 5

[5] Adobe Inc. Adobe photoshop. 3

[6] Adrian Albert, Jasleen Kaur, and Marta C Gonzalez. Using Convolutional Networks and Satellite Imagery to Identify Patterns in Urban Environments at a Large Scale. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1357–1366, 2017. 2

[7] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing Robust Adversarial Examples. In *International Conference on Machine Learning*, pages 284–293. PMLR, 2018. 2

[8] Lars Aurdal, Kristin Hammarstrøm Løkken, Runhild Aae Klausen, Alvin Brattli, and Hans Christian Palm. Adversarial camouflage for naval vessels. In *Artificial Intelligence and Machine Learning in Defense Applications*, volume 11169, pages 163–174. SPIE, 2019. 2

[9] Tao Bai, Jinqi Luo, and Jun Zhao. Inconspicuous Adversarial Patches for Fooling Image Recognition Systems on Mobile Devices. *IEEE Internet of Things Journal*, 9(12):9515–9524, 2021. 2

[10] Blender Cycles. https://docs.blender.org/manual/en/latest/render/cycles/index.html. Accessed: 2024-03-06. 4

[11] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial Patch. *arXiv preprint arXiv:1712.09665*, 2017. 2

[12] Yulong Cao, Ningfei Wang, Chaowei Xiao, Dawei Yang, Jin Fang, Ruigang Yang, Qi Alfred Chen, Mingyan Liu, and Bo Li. Invisible for both camera and lidar: Security of multi-sensor fusion based perception in autonomous driving under physical-world attacks. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 176–194. IEEE, 2021. 2

[13] Nicholas Carlini and David Wagner. Towards Evaluating the Robustness of Neural Networks. In *2017 IEEE Symposium on Security and Privacy*, pages 39–57. IEEE, 2017. 2

[14] Li Chen, Guowei Zhu, Qi Li, and Haifeng Li. Adversarial Example in Remote Sensing Image Recognition. *CoRR*, abs/1910.13222, 2019. 3

[15] Shang-Tse Chen, Cory Cornelius, Jason Martin, and Duen Horng Chau. Shapeshifter: Robust Physical Adversarial Attack on Faster R-CNN Object Detector. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2018, Dublin, Ireland, September 10–14, 2018, Proceedings, Part I 18*, pages 52–68. Springer, 2019. 2

[16] Ka-Ho Chow, Ling Liu, Margaret Loper, Juhyun Bae, Mehmet Emre Gursoy, Stacey Truex, Wenqi Wei, and Yanzhao Wu. Adversarial Objectness Gradient Attacks in Real-time Object Detection Systems. In *2020 Second IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, pages 263–272, 2020. 2

[17] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. 4

[18] Wojciech Czaja, Neil Fendley, Michael Pekala, Christopher Ratto, and I-Jeng Wang. Adversarial Examples in Remote Sensing. In *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 408–411, 2018. 3

[19] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, 2005. 2

[20] Deng, Jia and Dong, Wei and Socher, Richard and Li, Li-Jia and Kai Li and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 2

[21] Dosovitskiy, Alexey and Beyer, Lucas and Kolesnikov, Alexander and Weissenborn, Dirk and Zhai, Xiaohua and Unterthiner, Thomas and Dehghani, Mostafa and Minderer, Matthias and Heigold, Georg and Gelly, Sylvain and Uszkoreit, Jakob and Houlsby, Neil. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *ICLR*, 2021. 3

[22] Andrew Du, Bo Chen, Tat-Jun Chin, Yee Wei Law, Michele Sasdelli, Ramesh Rajasegaran, and Dillon Campbell. Physical Adversarial Attacks on an Aerial Imagery Object Detector. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1796–1806, January 2022. 3, 5, 6, 7

[23] Andrew Du, Yee Wei Law, Michele Sasdelli, Bo Chen, Ken Clarke, Michael Brown, and Tat-Jun Chin. Adversarial Attacks against a Satellite-borne Multispectral Cloud Detector. In *2022 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–8, 2022. 2

[24] Ranjie Duan, Xingjun Ma, Yisen Wang, James Bailey, A Kai Qin, and Yun Yang. Adversarial Camouflage: Hiding Physical-World Attacks with Natural Styles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1000–1008, 2020. 2

[25] Ranjie Duan, Xiaofeng Mao, A Kai Qin, Yuefeng Chen, Shaokai Ye, Yuan He, and Yun Yang. Adversarial Laser Beam: Effective Physical-World Attack to DNNs in a Blink. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16062–16071, 2021. 2

[26] Line Eikvil, Lars Aurdal, and Hans Koren. Classification-based vehicle detection in high-resolution satellite images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 64(1):65–72, 2009. 2

[27] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust Physical-World Attacks on Deep Learning Visual Classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[28] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust Physical-World Attacks on Deep Learning Visual Classification. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1625–1634, 2018. 2

[29] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Dawn Song, Tadayoshi Kohno, Amir Rahmati, Atul Prakash, and Florian Tramer. Note on Attacking Object Detectors with Adversarial Stickers. *arXiv preprint arXiv:1712.08062*, 2017. 2

[30] Kunihiko Fukushima. Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position. *Biological cybernetics*, 36(4):193–202, 1980. 2

[31] Ross Girshick. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015. 2

[32] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 2

[33] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 2

[34] Google LLC. Google maps. 3

[35] Rui Guo, Jasmine Collins, Oscar de Lima, and Andrew Owens. GANmouflage: 3D Object Nondetection with Texture Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4702–4712, 2023. 2

[36] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 2

[37] Huang, Lifeng and Gao, Chengying and Zhou, Yuyin and Xie, Cihang and Yuille, Alan L. and Zou, Changqing and

Liu, Ning. Universal Physical Camouflage Attacks on Object Detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 5

[38] Jiawei Lian and Shaohui Mei and Shun Zhang and Mingyang Ma. Benchmarking Adversarial Patch Against Aerial Detection. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–16, 2022. 2, 3

[39] Glenn Jocher. YOLOv5 by Ultralytics, May 2020. 7

[40] Yohei Koga, Hiroyuki Miyazaki, and Ryosuke Shibasaki. A Method for Vehicle Detection in High-Resolution Satellite Images that Uses a Region-Based Object Detector and Unsupervised Domain Adaptation. *Remote Sensing*, 12(3), 2020. 2

[41] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. 2

[42] Alexey Kurakin, Ian Goodfellow, Samy Bengio, et al. Adversarial Examples in the Physical World, 2016. 2

[43] S. Lazebnik, C. Schmid, and J. Ponce. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 2169–2178, 2006. 2

[44] Kibok Lee, Zhuoyuan Chen, Xinchen Yan, Raquel Urtasun, and Ersin Yumer. ShapeAdv: Generating Shape-Aware Adversarial 3D Point Clouds. *arXiv preprint arXiv:2005.11626*, 2020. 2

[45] Qizhang Li, Yiwen Guo, and Hao Chen. Practical No-Box Adversarial Attacks against DNNs. *Advances in Neural Information Processing Systems*, 33:12849–12860, 2020. 2

[46] Lin, Tsung-Yi and Goyal, Priya and Girshick, Ross and He, Kaiming and Dollar, Piotr. Focal Loss for Dense Object Detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2, 7

[47] LINZ - Selwyn 0.125m Urban Aerial Photos (2012-2013). https://data.linz.govt.nz/layer/51926-selwyn-0125m-urban-aerial-photos-2012-2013/. Accessed: 2024-03-06. 3

[48] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single Shot MultiBox Detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016. 2

[49] Xin Liu, Huanrui Yang, Ziwei Liu, Linghao Song, Hai Li, and Yiran Chen. DPatch: An Adversarial Patch Attack on Object Detectors. *arXiv preprint arXiv:1806.02299*, 2018. 2

[50] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into Transferable Adversarial Examples and Black-box Attacks. In *International Conference on Learning Representations*, 2017. 2

[51] Kristin Hammarstrøm Løkken, Lars Aurdal, Alvin Brattli, and Hans Christian Palm. Investigating robustness of adversarial camouflage (AC) for naval vessels. *Artificial Intelligence and Machine Learning in Defense Applications II*, 2020. 2

[52] D.G. Lowe. Object Recognition from Local Scale-Invariant Features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157 vol.2, 1999. 2

[53] Jiajun Lu, Hussein Sibai, and Evan Fabry. Adversarial Examples that Fool Detectors. *arXiv preprint arXiv:1712.02494*, 2017. 2

[54] Lu, Mingming and Li, Qi and Chen, Li and Li, Haifeng. Scale-Adaptive Adversarial Patch Attack for Remote Sensing Image Aircraft Detection. *Remote Sensing*, 13(20), 2021. 3

[55] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations*, 2018. 2

[56] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2

[57] Dario Pavllo, Jonas Kohler, Thomas Hofmann, and Aurelien Lucchi. Learning Generative Models of Textured 3D Meshes From Real-World Images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13879–13889, October 2021. 4

[58] Zirui Peng, Shaofeng Li, Guoxing Chen, Cheng Zhang, Haojin Zhu, and Minhui Xue. Fingerprinting Deep Neural Networks Globally via Universal Adversarial Perturbations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13430–13439, 2022. 2

[59] QGIS Development Team. *QGIS Geographic Information System*. QGIS Association, 2023. 3

[60] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3D Deep Learning with PyTorch3D. *arXiv:2007.08501*, 2020. 3, 4

[61] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2

[62] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. 2, 7

[63] Javier Ribera, David Guera, Yuhao Chen, and Edward J Delp. Locating Objects Without Bounding Boxes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6479–6489, 2019. 3

[64] Nataniel Ruiz, Adam Kortylewski, Weichao Qiu, Cihang Xie, Sarah Adel Bargal, Alan Yuille, and Stan Sclaroff. Simulated Adversarial Testing of Face Recognition Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4145–4155, 2022. 2

[65] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Yann Lecun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *Localization and Detection using Convolutional Networks*, pages 1–16, 01 2013. 2

[66] Michelle Shu, Chenxi Liu, Weichao Qiu, and Alan Yuille. Identifying Model Weakness with Adversarial Examiner. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11998–12006, 2020. 2

[67] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations*, 2015. 2

[68] Chawin Sitawarin, Arjun Nitin Bhagoji, Arsalan Mosenia, Mung Chiang, and Prateek Mittal. DARTS: Deceiving Autonomous Cars with Toxic Signs. *arXiv preprint arXiv:1802.06430*, 2018. 2

[69] Dawn Song, Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramer, Atul Prakash, and Tadayoshi Kohno. Physical Adversarial Examples for Object Detectors. In *12th USENIX Workshop On Offensive Technologies (WOOT 18)*, 2018. 2

[70] Delia-Georgiana Stuparu, Radu-Ioan Ciobanu, and Ciprian Dobre. Vehicle Detection in Overhead Satellite Images Using a One-Stage Object Detection Model. *Sensors*, 20(22), 2020. 2

[71] Huiming Sun, Jiacheng Guo, Zibo Meng, Tianyun Zhang, Jianwu Fang, Yuewei Lin, and Hongkai Yu. Evd4uav: An altitude-sensitive benchmark to evade vehicle detection in uav. *arXiv preprint arXiv:2403.05422*, 2024. 6, 7

[72] Suryanto, Naufal and Kim, Yongsu and Kang, Hyoeun and Larasati, Harashta Tatimma and Yun, Youngyeo and Le, Thi-Thu-Huong and Yang, Hunmin and Oh, Se-Yoon and Kim, Howon. DTA: Physical Camouflage Attacks Using Differentiable Transformation Network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15305–15314, June 2022. 2, 5, 6

[73] Suryanto, Naufal and Kim, Yongsu and Larasati, Harashta Tatimma and Kang, Hyoeun and Le, Thi-Thu-Huong and Hong, Yoonyoung and Yang, Hunmin and Oh, Se-Yoon and Kim, Howon. ACTIVE: Towards Highly Transferable 3D Physical Camouflage for Universal and Robust Vehicle Evasion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4305–4314, October 2023. 2, 5, 6

[74] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 1, 2

[75] Donghua Wang, Tingsong Jiang, Jialiang Sun, Weien Zhou, Zhiqiang Gong, Xiaoya Zhang, Wen Yao, and Xiaoqian Chen. FCA: Learning a 3D Full-coverage Vehicle Camouflage for Multi-view Physical Adversarial Attack. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 2414–2422, 2022. 2, 5, 6

[76] Derui Wang, Chaoran Li, Sheng Wen, Qing-Long Han, Surya Nepal, Xiangyu Zhang, and Yang Xiang. Daedalus: Breaking Nonmaximum Suppression in Object Detection via Adversarial Examples. *IEEE Transactions on Cybernetics*, pages 1–14, 2021. 2

[77] Jiakai Wang, Aishan Liu, Zixin Yin, Shunchang Liu, Shiyu Tang, and Xianglong Liu. Dual attention suppression attack: Generate adversarial camouflage in physical world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8565–8574, 2021. 5

[78] Zhen Wang, Buhong Wang, Yaohui Liu, and Jianxin Guo. Global Feature Attention Network: Addressing the Threat of Adversarial Attack for Aerial Image Semantic Segmentation. *Remote Sensing*, 15(5), 2023. 2

[79] Yuxin Wen, Jiehong Lin, Ke Chen, C. L. Philip Chen, and Kui Jia. Geometry-Aware Generation of Adversarial Point Clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(6):2984–2999, 2022. 2

[80] Chris Wise and Jo Plested. Developing Imperceptible Adversarial Patches to Camouflage Military Assets From Computer Vision Enabled Technologies. *arXiv preprint arXiv:2202.08892*, 2022. 3

[81] Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. DOTA: A Large-Scale Dataset for Object Detection in Aerial Images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[82] Chaowei Xiao, Dawei Yang, Bo Li, Jia Deng, and Mingyan Liu. MeshAdv: Adversarial Meshes for Visual Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6898–6907, 2019. 2

[83] Huangxinxin Xu, Fazhi He, Linkun Fan, and Junwei Bai. D3AdvM: A direct 3D adversarial sample attack inside mesh data. *Computer Aided Geometric Design*, 97:102122, 2022. 2

[84] Yonghao Xu and Pedram Ghamisi. Universal Adversarial Examples in Remote Sensing: Methodology and Benchmark. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–15, 2022. 3

[85] Zhaoxia Yin, Qingyuan Wang, Jin Tang, and Bin Luo. Universal adversarial perturbation for remote sensing images. *2022 IEEE 24th International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6, 2022. 3

[86] Tao Yu, Runseng Feng, Ruoyu Feng, Jinming Liu, Xin Jin, Wenjun Zeng, and Zhibo Chen. Inpaint Anything: Segment Anything Meets Image Inpainting. *arXiv preprint arXiv:2304.06790*, 2023. 3

[87] Xiaohui Yuan, Jianfang Shi, and Lichuan Gu. A review of deep learning methods for semantic segmentation of remote sensing imagery. *Expert Systems with Applications*, 169:114417, 2021. 2

[88] Zhang, Yichuang and Zhang, Yu and Qi, Jiahao and Bin, Kangcheng and Wen, Hao and Tong, Xunqian and Zhong, Ping. Adversarial Patch Attack on Multi-Scale Object Detection for UAV Remote Sensing Images. *Remote Sensing*, 14(21), 2022. 3

[89] Xingkui Zhu, Shuchang Lyu, Xu Wang, and Qi Zhao. TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Sce-

narios. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2778–2788, 2021. 2