Exploit, Defend, and Detect Adversarial Attacks on Image-Based AI Systems: An Ethical Perspective

Gualandi Mattia - mattia.gualandi2@studio.unibo.it
Presepi Alex - alex.presepi@studio.unibo.it
Sciarrillo Alessandro - alessandr.sciarrill2@studio.unibo.it

August 26, 2025

Abstract

The growing integration of deep learning models in critical domains has brought urgent attention to their vulnerabilities. One of the most concerning threats is adversarial attacks on image-based AI systems: subtle perturbations to input images that can mislead even the most accurate models into making incorrect predictions, often without detection.

These attacks challenge not only the technical robustness of AI models but also raise significant ethical concerns, particularly when human safety, fairness, and accountability are at stake.

In this project, we explore the nature of adversarial attacks on image data, aiming to deepen our understanding of their mechanisms and implications. We review and categorize popular attack techniques, evaluate their impact on standard vision models, and assess the effectiveness of different models we designed and trained to counteract the considered attacks.

Furthermore, we developed and tested different versions of a module capable of recognizing when an input image has been corrupted. Finally, the analysis is extended to practical case studies in face recognition and vehicle detection, where we investigate how adversarial attacks can compromise security and fairness in real-world applications. By framing the problem from both a technical and ethical perspective, we highlight the importance of designing AI systems that are not only performant but also resilient and trustworthy under adversarial conditions.

1 Introduction

The increasing reliance on deep learning models in critical applications has revolutionized how we approach tasks such as medical diagnostics, autonomous driving, and security surveillance. However, as these models become embedded in real-world systems, their vulnerabilities have drawn growing scrutiny. Among the most notable and alarming of these weaknesses are adversarial attacks: small, carefully crafted perturbations to input images that can mislead even state-of-the-art models into making incorrect predictions, often without raising any obvious signs, as shown in Figure 1. These attacks not only expose technical limitations of deep learning architectures but also undermine trust in AI systems, particularly in scenarios where human safety and accountability are paramount.

In this work, we first focused on thoroughly exploring the theoretical foundations and effects of adversarial attacks and defenses within a simplified digital scenario. This initial step allowed us to understand the core mechanisms in a controlled setting. We then moved to the practical side of the problem, applying theory to real-world contexts. Our goal was twofold: to highlight the concrete risks adversarial attacks pose in practice, such as during physical penetration tests in cybersecurity, and to evaluate how the studied defenses could effectively mitigate these threats in real situations.

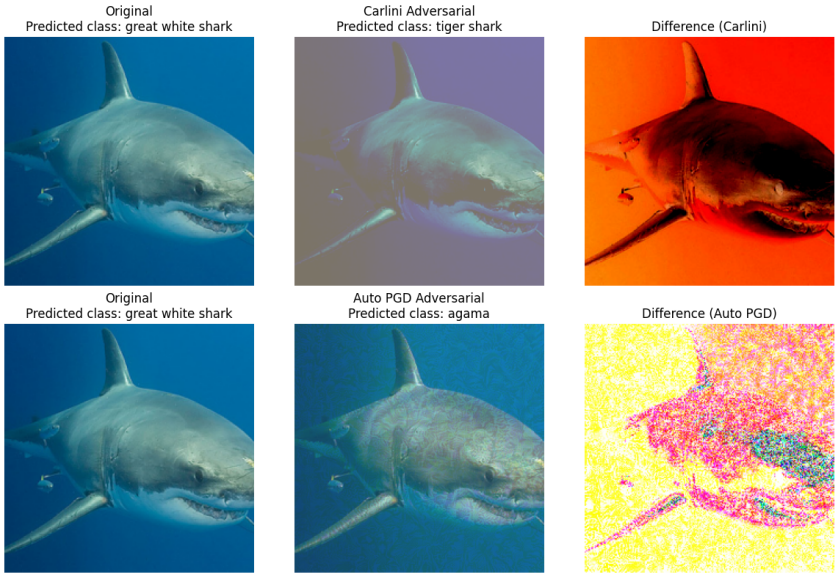

Figure 1: Neural network fooling with Carlini and PGD attacks.

2 Motivation

The motivation for this project stems from the urgent need to understand and mitigate the risks posed by adversarial attacks in image-based AI systems. As these attacks can compromise the reliability of models deployed in sensitive domains, addressing them is crucial to ensuring both the functional robustness and the ethical integrity of AI solutions. By investigating the mechanics and consequences of different attack strategies, we aim to uncover insights into their effectiveness against standard vision models and develop defenses that are resilient, trustworthy, and transparent. Beyond the technical dimension, the project is also motivated by the broader societal implications of adversarial vulnerabilities—highlighting how weaknesses in AI can jeopardize fairness, accountability, and ultimately public confidence in automated systems. In particular, we focus on two real-world scenarios: face recognition and automatic surveillance systems to demonstrate how seemingly minor perturbations can pose a direct threat to privacy, security, and fairness.

3 Related Work

Research on adversarial attacks and defenses has grown rapidly since the early discoveries of vulnerabilities in deep neural networks [11, 5]. A wide range of attack techniques have been proposed, including gradient-based methods such as FGSM and PGD [6], optimization-based approaches like the Carlini & Wagner (C&W) attack [1], and decision-

based methods such as HopSkipJump [2]. Other notable contributions include DeepFool [7], the one-pixel attack [10], and evaluation frameworks like AutoAttack [3]. On the defense side, adversarial training [6, 13], alongside techniques such as input preprocessing [9], detection methods [12], have been tried to assess the improvement margin of commonly employed models. Recent surveys have highlighted not only the technical complexity of designing robust models but also the broader ethical and safety concerns surrounding adversarial attacks [15]. Building on this foundation, our work systematically evaluates prominent attack methods, investigates adversarial training to enhance robustness, and introduces a detection module for adversarially perturbed images, considering both technical effectiveness and ethical implications.

4 Methods

In this section, we describe the main techniques studied in this project. We focus on the different ways an attacker can generate adversarial examples, the strategies that can be employed to defend against them, and the tools available to measure model robustness. We distinguish between white-box attacks (where the attacker has full knowledge of the model architecture, parameters, and gradients) and black-box attacks (where the attacker can only observe model outputs, such as predicted labels or probabilities), and finally describe the defences, detection approaches, and robustness metrics.

An important distinction when discussing adversarial attacks is between targeted and untargeted variants. In a targeted attack, the goal is to modify an input so that the model classifies it as a specific target class chosen by the attacker. This typically requires more precise optimization, as the perturbation must move the input representation toward the decision boundary of the chosen class. In contrast, an untargeted attack only aims to cause any misclassification, without specifying which incorrect class is predicted. Untargeted attacks are often easier to perform and can require smaller perturbations to succeed. Some attack algorithms are designed to support both targeted and untargeted modes (e.g., PGD, C&W), while others are inherently formulated for just one of the two settings.

In our experiments, we tested the selected attacks on the ImageNet100 dataset, adjusting parameters to explore their most efficient configurations. We employed both targeted and untargeted versions exclusively for PGD, in order to compare their relative effectiveness. For all other attack methods considered in this work, we opted for the untargeted variant, as it represents the more general and widely applicable threat scenario.

Moreover, we test defences in terms of accuracy improvement and generalization, together with the detector module described above.

4.1 White-Box Attacks

White-box attacks assume that the adversary has complete knowledge of the target model. This includes knowing the model architecture, the learned parameters (weights), and having the ability to compute gradients with respect to inputs. These attacks are powerful because they directly exploit the way the network processes information.

Projected Gradient Descent (PGD)

Projected Gradient Descent (PGD) [6] is one of the most widely used and influential white-box attack methods. The key idea is that small changes to an image, if carefully chosen, can cause a model to misclassify it. PGD works iteratively: starting from the original image, it calculates the gradient of the model's loss with respect to the image (essentially showing how to change the pixels to make the model more wrong). After each small change, the image is "projected" back into a constrained region around the original image (called an ℓ_p-ball) so that the final adversarial image looks very similar to the original. By repeating this process many times, PGD finds perturbations that are very effective but remain imperceptible. PGD is considered a "universal first-order adversary" and is frequently used as a benchmark attack in research.

Auto-PGD (APGD)

Auto-PGD [3] is an improved version of PGD that automatically adjusts its parameters during the attack. One of the practical challenges of PGD is deciding the size of each gradient step and the number of steps to take. APGD removes this manual tuning by using adaptive step sizes and backtracking: if a step is too large and does not help, it automatically reduces it. It also introduces a momentum-based approach that makes it more stable and efficient. This results in a more reliable attack that works well across different models without manual configuration.

Carlini & Wagner (C&W) L2 Attack

The Carlini & Wagner attack [1] formulates adversarial sample generation as an optimization problem. Instead of applying simple gradient steps, this attack explicitly searches for the smallest possible change to the image (measured with the ℓ_2 norm) that will fool the network. It does so by defining a custom objective function that penalizes large perturbations and uses advanced optimization techniques to minimize this function. The result is an image that looks almost identical to the original but causes the model to make an incorrect prediction. This attack is known to be extremely effective, especially against models that attempt to hide their gradients (a defense called gradient masking).

Feature Adversaries

Feature adversaries [9] target not the final classification output but the internal feature representations of the network. Deep networks process an image through several layers, extracting progressively more abstract features (edges, shapes, patterns). This attack forces the features of an image to resemble those of another target image by introducing small perturbations. As a result, the model is confused and produces incorrect predictions. These attacks highlight that adversarial vulnerabilities are present not just at the output stage, but also in the way the network processes information internally.

Shadow Attack

The Shadow Attack [4] introduces a novel class of adversarial examples that exploit the interplay between classifier outputs and their associated robustness certificates. Unlike traditional adversarial attacks that focus solely on misclassification, the Shadow Attack aims to deceive certified classifiers by generating perturbations that are large in ℓ_p-norm,

yet visually imperceptible. These perturbations cause the classifier to mislabel the input while simultaneously producing a "spoofed" certificate of robustness, falsely indicating that the perturbed image is not adversarial within a specified ℓ_p-ball. This dual deception undermines the reliability of certified defenses, highlighting that such systems can be vulnerable to attacks that manipulate both the classifier's decision and its confidence guarantees [4].

DeepFool

The DeepFool attack [7] is an efficient method designed to compute minimal perturbations that cause deep neural networks to misclassify input samples. The algorithm operates by approximating the decision boundary of the classifier using linear classifiers and iteratively finding the smallest perturbation that moves the input sample across this boundary. Specifically, for a given input image, DeepFool computes the perturbation by linearizing the classifier's decision function around the current input and determining the direction and magnitude of the perturbation needed to cross the decision boundary. This process is repeated until the input is misclassified. DeepFool has been shown to outperform existing methods in terms of the size of the perturbations and the number of iterations required to achieve misclassification, making it a valuable tool for evaluating the robustness of deep neural networks to adversarial attacks [7].

4.2 Black-Box Attacks

In black-box attacks, the attacker does not know the internal details of the model. They can only observe the model's outputs (predicted labels or scores) when they submit inputs. These attacks rely on exploring the input space through queries, often requiring more attempts but demonstrating that even limited access can be dangerous.

HopSkipJump Attack

The HopSkipJump attack [2] is a decision-based black-box attack. It does not need gradients or confidence scores; it only requires the model's final decisions (for example, "cat" or "dog"). The attack works by starting from an input that is already misclassified and gradually moving towards the original image while staying close to the model's decision boundary. It estimates the direction to move by trying small changes and observing whether the label changes. Over many iterations, this process produces an adversarial example that is very similar to the original.

One Pixel Attack

The One Pixel Attack [10] is a striking example of how sensitive deep models can be. It shows that changing just a single pixel in an image can sometimes be enough to fool a network. The method uses an evolutionary algorithm (a type of optimization inspired by natural selection) to find the position and value of the pixel (or a very small set of pixels) that will cause the misclassification. Even though this attack requires many queries, it demonstrates that not all adversarial attacks need large or complex changes.

4.3 Defences

Defences are techniques that aim to make models less vulnerable to adversarial attacks. They can be divided into strategies that change the model's behaviour (e.g., during training) and those that modify the inputs before feeding them into the model.

Spatial Smoothing

Spatial smoothing is a simple, preprocessing-based defense. Before passing the image to the model, a filter (such as a Gaussian blur or median filter) is applied. This filter removes high-frequency noise, which often includes adversarial perturbations, while keeping the main content of the image. Although it can mitigate some attacks, strong adaptive attacks can usually circumvent it, so it is considered a basic defense.

Feature Squeezing Preprocessing

Feature squeezing is a lightweight input-transform defense aimed at detecting adversarial examples by reducing the adversary's degrees of freedom in the input space [14]. The core idea is to apply simple input transformations, such as reducing the color bit depth of each pixel or applying spatial smoothing, and then compare the model's original prediction with the one obtained from the "squeezed" input. A large discrepancy between predictions (e.g., measured via L_1 distance) indicates a likely adversarial input. This method has shown high detection rates across multiple datasets while maintaining accuracy on benign inputs, and can also be composed with other defenses such as adversarial training [14].

Adversarial Training

Adversarial training [6] is a much stronger defense. The idea is to expose the model to adversarial examples during training so that it learns to resist them. During the training, adversarial examples are incorporated into the training dataset. Over time, the model learns parameters that reduce the effect of these perturbations. While this greatly improves robustness, it is computationally expensive because it requires generating adversarial examples, possibly for a large number of samples, in order to be effective. The main drawback of adversarial training is that its effectiveness strongly depends on the adversarial samples used during training. In many cases, the resulting model becomes robust primarily to the specific type of attack it was trained against, offering limited generalization to other attack methods. This lack of transferability can be restrictive in practical scenarios. On the other hand, certain attack strategies used during training can induce broader robustness, improving the model's resistance to a wider range of adversarial perturbations.

Fast Is Better Than Free (FBF)

Fast Is Better Than Free [13] is a more efficient variation of adversarial training. Instead of using slow, multi-step attacks like PGD, it uses a single-step attack combined with random initialization. A special training schedule (cyclic learning rate) is used to maintain robustness while keeping training time manageable. This makes it feasible to apply adversarial training on large datasets and models. In our case, being the use of PGD and other attacks not that prohibitive in terms of computational load, this approach

has been just tried for the sake of completeness, but the results, possibly due to the more sophisticated nature of our models, with respect to the simpler one used in the original paper [13], resulted in a very unsatisfying performance, which was not worth to be mentioned later.

4.4 Detector

Adversarial Detector

The adversarial detector operates as an auxiliary module whose task is to determine whether an incoming sample is benign or adversarially perturbed before it is processed by the primary classifier. Technically, we distinguished between two detectors: the input and the activation detectors.

The first one consists basically in a trained binary classification head attached to the existing classifier, kept frozen; while the second is a module, arbitrarily complicated, attached to the backbone and receives as input an intermediate activation of the main classifier, as shown in Figure 2. In both cases, the detector produces an output that can be used as a continuous anomaly score (e.g., the probability of being adversarial), or directly a binary label (clean/adversarial) by thresholding the score. The output can trigger one of several responses: for instance, (i) allow normal classification, (ii) reject and flag for manual review, or (iii) route the sample through a hardened processing pipeline.

Formally, let x be the input sample, f_θ the main classifier, and d_ϕ the detector network. The detector computes:

$$y_d = d_\phi(x) \quad \text{or} \quad y_d = d_\phi(h_l(x)),$$

where $h_l(x)$ denotes the feature representation at layer l of f_θ. The output y_d can be used as it is or compared to a threshold τ to decide whether the sample is adversarial:

$$\text{Adversarial if } y_d > \tau.$$

This modular design allows the detector to function independently from the classifier's predicted class, which is crucial for detecting attacks that do not necessarily force a specific misclassification.

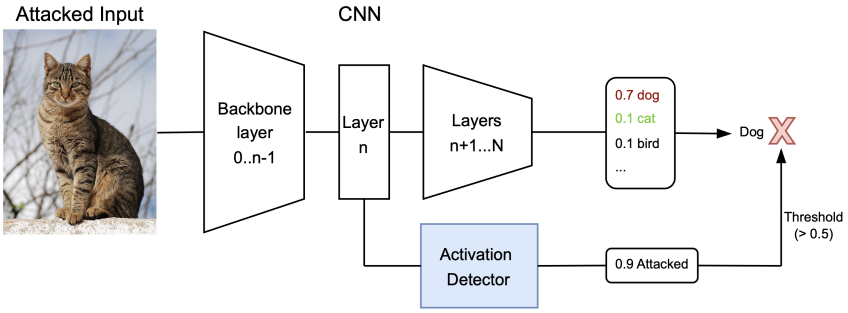

Figure 2: Activation detector schema drawn by us for the sake of clarity.

4.5 Metrics

Attack Evaluation Metrics

When evaluating adversarial attacks, it is important to quantify the magnitude of the perturbations and the effectiveness of the attack in degrading the model's performance. A commonly used family of metrics is based on the ℓ_n norms of the perturbations; moreover, we will compare the accuracy drop of the model on the adversarial dataset.

ℓ_p Norms The ℓ_p norm of a perturbation measures its size in a mathematical sense, providing a way to compare the strength or subtlety of different attacks. For an adversarial perturbation vector δ, the ℓ_p norm is defined as:

$$\|\delta\|_p = \left(\sum_{i=1}^{d} |\delta_i|^p \right)^{1/p} \quad \text{for} \quad 1 \leq p < \infty$$

and for $p = \infty$:

$$\|\delta\|_\infty = \max_i |\delta_i|$$

Common choices include:

- ℓ_1: Measures the sum of absolute differences, often linked to sparsity of perturbations.
- ℓ_2: Measures the Euclidean distance between the clean and perturbed samples, capturing overall perturbation energy.
- ℓ_∞: Measures the maximum change to any single input component (pixel), often used in image-based attacks to constrain per-pixel changes.

Additionally, just for the object detection scenario, we use the following:

Mean Average Precision (mAP) and Mean Average Recall (mAR) In detection-based evaluation, *mean average precision* (mAP) and *mean average recall* (mAR) are standard measures. For C classes and n recall or precision levels:

$$\text{mAP} = \frac{1}{C} \sum_{c=1}^{C} \left(\frac{1}{n} \sum_{i=1}^{n} P_i \right), \quad \text{mAR} = \frac{1}{C} \sum_{c=1}^{C} \left(\frac{1}{n} \sum_{i=1}^{n} R_i \right)$$

where precision P and recall R are defined as:

$$P = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad R = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Here, TP, FP, and FN denote true positives, false positives, and false negatives, respectively.

Attack Success Rate (ASR) The *Attack Success Rate* [15]quantifies the effectiveness of adversarial perturbations by measuring the degradation in performance due to the attack. It is defined as:

$$\text{ASR} = 1 - \frac{M_{\text{attack}}}{M_{\text{clean}}}$$

where M_{attack} and M_{clean} are the values of a chosen evaluation metric (e.g., mAP or mAR) computed on the adversarially perturbed dataset and the clean dataset, respectively. A higher ASR indicates that the attack has caused a more significant drop in model performance. This definition directly ties the success of the attack to its impact on the adopted task-specific performance metric.

Robustness Metric: CLEVER

The CLEVER score [12] is a way to measure how resistant a model is to small changes in its input. Instead of testing the model with specific attacks, CLEVER looks at how sensitive the model is in general. It does this by adding many tiny random changes to the input and checking how much the output varies. Using statistics, it then estimates how large a change would be needed to actually alter the model's decision. A higher CLEVER score means that bigger changes are required, so the model is considered more robust.

Experimental Setup

All experiments in this section are conducted using the ResNet18 backbone model and the ImageNet100 dataset, both of which are widely adopted in research and practice. In particular, we fine-tuned and tested ResNet18 on ImageNet100 in order to analyze adversarial robustness with models and datasets that are representative of real-world applications. This choice is especially relevant, as it demonstrates that fragility is not limited to toy models, often used in preliminary studies, but also affects more advanced architectures that are extensively employed in real systems, such as ResNet. The ImageNet100 dataset is constructed by selecting 100 classes from the 1,000 available in ImageNet, ensuring a balanced distribution across categories while significantly reducing the overall size. This makes it a practical choice for adversarial training and evaluation, as it preserves the diversity and complexity of ImageNet while allowing for faster experimentation and manageable computational costs.

In analyzing the adversarial attacks used in our study, it is crucial to consider the balance between effectiveness and computational efficiency. Some methods, such as DeepFool, Feature Adversaries, and black-box approaches like HopSkipJump or Pixel Attack, proved to be particularly time-consuming due to their iterative nature and reliance on gradient estimation or fine-grained feature optimization. In contrast, white-box attacks such as PGD or Auto-PGD offer a better compromise between strength and efficiency, as they can be executed faster and with more predictable runtime. For each attack, we selected hyperparameters, such as step size, perturbation bounds, and number of iterations, by carefully balancing robustness of evaluation, computational feasibility, and samples' quality.

In cases where runtime became prohibitive (e.g., for black-box or feature-level attacks), we reduced the number of samples or iterations to ensure tractability while preserving the possibility of getting an approximate evaluation. To further investigate hyperparameter

sensitivity, we performed two grid searches on PGD and C&W attacks on the parameter we noticed were the most influential, visually comparing adversarial examples generated under different parameter settings, as reported in Table 1 and 2, and showed in Figure 3 and 4. In particular, for the PGD attack, the epsilon step controls how much the input is changed at each iteration, with larger values leading to stronger but less precise perturbations. For the C&W attack, the confidence parameter determines how strongly the attack pushes the model to misclassify: higher confidence produces adversarial samples that are harder to defend against but also more visibly altered.

Overall, this analysis underlines the trade-off between attack strength, stealth, and computational practicality across different adversarial strategies.

Epsilon step \ Metrics	ℓ_0	ℓ_1	ℓ_2	ℓ_∞
1	0.85	3.10	3.33	0.03
0.1	0.77	1.54	1.60	0.02
0.01	0.73	1.23	1.25	0.01
0.001	0.73	1.22	1.25	0.01

Table 1: Grid search on epsilon step parameter on PGD attack.

Confidence \ Metrics	ℓ_0	ℓ_1	ℓ_2	ℓ_∞
0.0	0.36	0.50	0.65	0.01
0.1	0.33	0.47	0.60	0.01
0.5	0.31	0.44	0.57	0.01
1.0	0.28	0.41	0.53	0.01

Table 2: Grid search on confidence parameter on C&W attack.

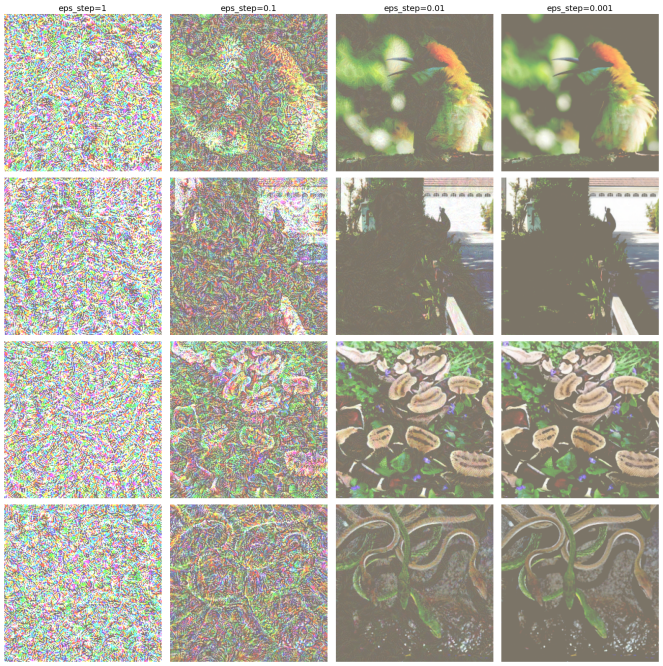

Figure 3: Grid search visualization results for PGD.

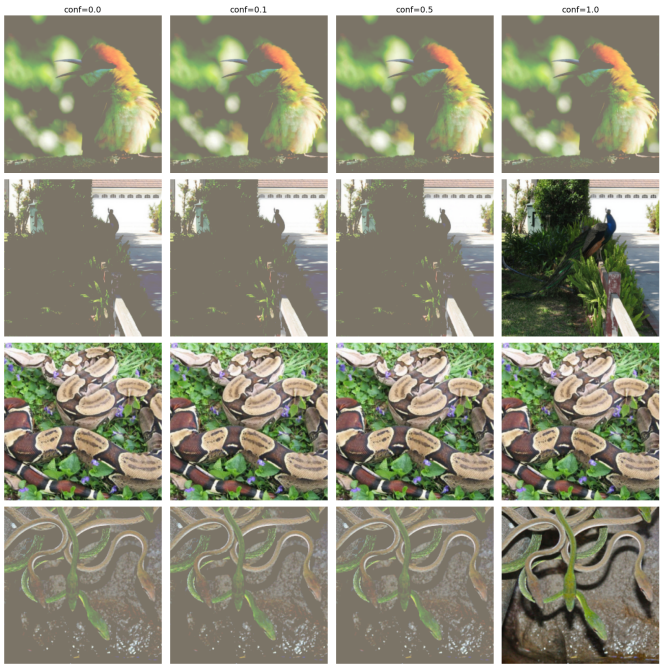

Figure 4: Grid search visualization results for C&W.

From the figures above, we observe that PGD is highly sensitive to the choice of epsilon. For example, with larger epsilon step values, the generated samples become almost unrecognizable to the human eye, whereas smaller values produce much more natural-looking results. In contrast, the Carlini attack maintains a more consistent sample quality across different settings, which is in line with the nature of this method.

Finally, we report the hyperparameters used for each attack in Table 3.

Consider that, as anticipated above, the whole dataset has been adversarially transformed except for some computationally expensive attacks, in particular: HSJ (300 samples), Pixel Attack (300 samples), Feature Adversaries (1000 samples), DeepFool (1000 samples).

	Hyperparameters
PGD Targeted	max_iter=20; eps_step=0.001; eps=5; decay=0.9
PGD Untargeted	max_iter=20; eps_step=0.001; eps=5; decay=0.9
Auto-PGD	eps=0.3; eps_step=0.1; max_iter=100, nb_rand_init=5; loss=CE
C&W	conf=0.0; lr=0.01; binary_search_steps=10; max_iter=10; init_const=0.01; max_halving=5; max_doubling=5
Feature Adversaries	delta=1.0; lambda=0.0; layer=-1; max_iter=25; step_size=0.01
Shadow Attack	sigma=0.01; nb_steps=300; lr=0.2; lambda_tv=0.3; lambda_c=1.0; lambda_s=0.5
DeepFool	eps=10^{-6}; max_iter=100; ng_grads=20
HopSkipJump	norm=2; max_iter=25; max_eval=10000; init_eval=10
Pixel Attack	es=1; max_iter=100

Table 3: Attacks' hyperameters used for adversarial sample creation.

Experimental Results

Attack Results

We begin by evaluating how the different attacks perform against the baseline model. Table 4 reports the perturbation percentages under the ℓ_n norms, together with the classification accuracy of the plain model.

	ℓ_0	ℓ_1	ℓ_2	ℓ_∞	Model Accuracy
PGD Targeted	0.39	0.54	0.69	0.01	0.07
PGD Untargeted	0.39	0.54	0.69	0.01	0.008
Auto-PGD	0.01	0.28	0.25	0.01	0.009
C&W	0.34	0.48	0.62	0.01	0.12
Feature Adversaries	0.74	1.20	1.22	0.01	0.01
Shadow Attack	0.74	1.28	1.37	0.02	0.0
DeepFool	0.75	1.23	1.26	0.03	0.01
HopSkipJump	0.76	1.21	1.23	0.02	0.01
Pixel Attack	0.01	0.01	0.87	0.92	0.01

Table 4: Metrics for different attacks on the baseline model.

Overall, it can be observed that the first group of attacks (the PGD family and C&W) typically produces subtle perturbations with lower norm values, making them more stealthy and generally harder to be noticed from a human as well, while still significantly reducing model accuracy. In contrast, optimization-based and black-box methods generally succeed in causing misclassification but at the cost of larger perturbations in both ℓ_1 and ℓ_2 norms, which makes them less inconspicuous. The notable exception is the Pixel Attack: by altering only a single pixel, it achieves very low perturbation across most measures, with the exception of ℓ_∞, which captures the maximum local distortion rather than the overall change.

This points to a clear trade-off between imperceptibility and distortion: attacks with smaller perturbation norms tend to be less noticeable but usually require stronger access to the model, while those with higher distortions can succeed in more constrained settings, though they leave more evident traces on the input.

Taken together with their lower computational cost, these findings suggest that the first four attacks represent the most practical choice for evaluation in this context, offering a balanced compromise between efficiency and distortion.

Defense Results

We next evaluate the performance of different defense strategies.

Pre-processing Defences We begin with simple preprocessing defenses that do not alter the model itself, namely spatial smoothing and feature squeezing. Their effectiveness is tested, for the sake of simplicity, on adversarial samples created with a targeted PGD attack, which was one of the seemingly most subtle attacks considering Table 4 results. Preprocessing defences results are presented in Table 5.

	Original without Defence	Original with Defence	Adversarial without Defence	Adversarial with Defence
Spatial Smoothing	0.67	0.60	0.07	0.18
Feature Squeezing	0.67	0.66	0.07	0.25

Table 5: Model accuracy with and without simple defenses on original and adversarial datasets.

As it could be expected, simple preprocessing defences are not that effective on the adversarial dataset.

Adversarial Training We then analyze adversarial training by training separate models on adversarial examples generated from individual attacks. Each model is tested on two datasets: (i) the original validation dataset augmented with adversarial validation examples from the same attack used for training, and (ii) a dataset containing adversarial examples from all attacks. This setup enables us to evaluate both attack-specific robustness and the generalization of robustness to unseen attacks. Results are reported in Table 6 and Table 7. The accuracy achieved by the baseline model on the original validation set is 0.67.

Adversarial Model	Original val-dataset	Single adversarial dataset	CLEVER
PGD Targeted	0.72	0.57	0.61
PGD Untargeted	0.71	0.50	0.83
Auto-PGD	0.68	0.61	1.29
C&W	0.73	0.63	0.38
Feature Adversaries	0.73	0.01	0.64
Shadow Attack	0.71	0.52	0.35
DeepFool	0.76	0.005	0.69
HopSkipJump	0.65	0.017	0.58
Pixel Attack	0.73	0.033	1.15

Table 6: Results of adversarial training with different attacks.

The results show that the accuracy on the original dataset remains fairly similar across different attacks, and in most cases even improves compared to the baseline. This improvement is likely due to the additional training epochs on an augmented dataset containing both original and adversarial images.

In contrast, performance on the adversarial dataset varies widely. While some attacks allow the model to maintain reasonable accuracy, others lead to almost complete failure, with the model unable to classify correctly at all. This suggests that certain attacks produce perturbations that are either too disruptive or too subtle to be effectively learned. Another factor to consider is the varying number of adversarial samples generated by different attacks: smaller sample sizes naturally limit generalization, although not enough to fully explain the observed performance gap.

These findings highlight the importance of having lightweight attacks capable of generating large adversarial datasets. Such datasets are essential for thoroughly training models to improve robustness, even though achieving satisfactory robustness remains a

challenging goal.

Moreover, CLEVER score, explained in Section 4.5, is computed on a small subset due to its computational cost and, from the obtained results, it partially mirrors the results of the adversarial training: in fact, while being reasonably decent for good performing models such as PGD based ones, it is surprisingly high also for models that seem not to achieve robustness, being them inaccurate also on the dataset they have been trained on, such as DeepFool or PixelAttack. This may be linked to the fact that, particularly hard attacks, due to too large (DeepFool) or too small (PixelAttack) perturbations, are still able to make models more robust according to CLEVER score when used for adversarial training.

Model\Dataset	PGD-T	PGD-U	Auto-PGD	C&W	Feature Adv.	Shadow	DeepFool	HSJ	Pixel	Mean
PGD-T	0.57	0.40	0.01	0.62	0.01	0.03	0.0	0.03	0.0	0.19
PGD-U	0.58	0.50	0.07	0.59	0.01	0.02	0.0	0.03	0.0	0.2
Auto-PGD	0.30	0.27	0.61	0.32	0.01	0.01	0.0	0.0	0.01	0.17
C&W	0.44	0.13	0.01	0.63	0.0	0.05	0.0	0.05	0.0	0.15
Feature Adv.	0.14	0.01	0.01	0.31	0.01	0.01	0.0	0.01	0.0	0.05
Shadow Attack	0.32	0.07	0.01	0.52	0.01	0.52	0.0	0.05	0.01	0.17
DeepFool	0.18	0.02	0.01	0.38	0.0	0.01	0.0	0.03	0.0	0.07
HopSkipJump	0.14	0.01	0.01	0.37	0.0	0.02	0.01	0.01	0.01	0.07
Pixel Attack	0.25	0.10	0.01	0.30	0.01	0.02	0.0	0.03	0.03	0.08

Table 7: Cross-evaluation of adversarial training (rows) against adversarial test sets (columns). The "Mean" column shows the average accuracy across all attacks.

Now, we observe the cross evaluation among adversarially trained models on all the datasets. Note that the elements on the diagonal are the same as the second column of Table 6. Considering the results, it can be noticed how some attacks are undefeatable in this framework, where none of the models, often neither the one trained on that dataset itself (elements belonging to the diagonal), is able to achieve decent accuracy. Some possible causes may be linked to a too big distortion or a too subtle attack, and this will be assessed later with detector evaluation and visualization.

If we exclude the dataset on which the average accuracy is basically 0, considering the other ones, we can observe that PGD-based trained models and C&W ones are the most transferable ones, achieving a decent accuracy not only on the samples they have been trained on but also on differently attacked ones. This may be caused by the nature of the attacks themselves, that, by design are not too invasive visually but still generally homogeneous. For this reason, these methods will also be the ones that will be used to assess the later analyzed real-world scenario.

Additionally, looking at the problem from a different perspective, it can be noticed that C&W and targeted PGD are the easiest detectable attacks, being them decently tackled from a set of models.

Detectors Finally, we assess the performance of a dedicated detector. The detector is tuned using the adversarial dataset with the C&W attack. In the first place, in order to explore its effectiveness, we test the activation detector at different integration points within the network: after the first conv2, after the first Resnet's stage, after the third, and after the fourth one. The corresponding results are shown in Table 8. This analysis provides insights into how the choice of input representation affects detection performance. Then the best activation detector and the input detector are then evaluated on a dataset composed of both adversarial samples from all attacks and the original clean images, reported in Table 9.

	First Conv2	ResNet Stage 1	ResNet Stage 3	ResNet Stage 5
Accuracy	**0.96**	0.95	0.93	0.94

Table 8: Activation detectors accuracy when attached at different layers of the network on the validation set.

Observing the results, the hypothesis behind the decreasing scores from the early layers to the later ones is that the effect of the attack tends to disperse as it propagates through the network stages, especially for very stealthy attacks. As a result, the best detection scores are obtained when observing the activations in the earlier stages, in this case the first convolutional layer, since all the spatial features are still present and not yet degraded.

Dataset\Detector	Input Detector	First Conv2 Activation Detector
PGD-T	0.96	**0.99**
PGD-U	0.96	**0.99**
Auto-PGD	**0.57**	0.51
C&W	0.93	**0.96**
Feature Adv.	0.95	**1.00**
Shadow Attack	0.97	**0.99**
DeepFool	0.97	**1.00**
HopSkipJump	0.97	**0.99**
Pixel Attack	0.47	**0.69**

Table 9: Input vs best activation detector on all attacks.

From this second table, we observe that, even if both detectors perform very well, the strongest performance comes from analyzing the very early activations, while the Input detector, which had access to an entire pretrained backbone for feature extraction, was outscored. In essence, a raw embedding from the first layers is more effective for spotting an attack than a refined embedding produced by the whole backbone.

By combining insights from attack evaluation, adversarial training, and detector performance, Table 10 summarizes the perturbation in ℓ_1 norm, the best accuracy achieved by a model not trained on the considered attack (best cross-model), and the accuracy of the best detector configuration. Additionally, the visualizations in Figure 5 help illustrate the relationship between attack quality and the effectiveness of the model's defenses.

Attack	Perturbation (ℓ_1)	Best Cross Model Accuracy	Best Detector Accuracy
PGD Targeted	0.54	0.58	0.99
PGD Untargeted	0.54	0.40	0.99
Auto-PGD	0.28	0.07	0.57
C&W	0.48	0.62	0.96
Feature Adversaries	1.20	0.01	1.00
Shadow Attack	1.28	0.05	0.99
DeepFool	1.23	0.01	1.00
HopSkipJump	1.21	0.05	0.99
Pixel Attack	0.01	0.01	0.69

Table 10: Comparison of attacks in terms of input perturbation, best cross-model accuracy under attack, and best detector accuracy.

In particular, attacks that cause the most problems in the adversarial training case but have high perturbation percentages, are generally easily predictable with the detector module, which still performs good also on other datasets, while some other, even if seemingly more subtle in terms of perturbations, give better results in adversarial training.

The most challenging attacks are Auto-PGD and Pixel Attack, whose adversarial scores are very low, none of the adversarially trained model can generalize (neither the one trained on the respective datasets themselves), and also their detection score is lower than the others; making their samples hard to manage due to good quality and stealthiness.

Finally, we can state that correct classification and generalization are, in general, difficult, especially among very different attack types, while detection proved to be a much more reliable task and a very good candidate to be a reliable countermeasure in real-world applications.

Still, it is evident how merging different defences can highly improve performance and reliability in avoiding adversarial samples from being treated as clean.

Figure 5: Attacked samples comparison.

5 Real-World Scenario: From Theory to Practice

In recent years, research on adversarial attacks has advanced considerably, producing a wide range of methods capable of fooling even state-of-the-art deep learning models in controlled environments. While these advances have enriched our theoretical understanding of model vulnerabilities, it is equally important to investigate how these threats manifest in real-world applications. Bridging the gap between theory and practice is essential in cybersecurity and AI safety, as the ultimate objective is to ensure that systems deployed in the field remain trustworthy and resilient against malicious manipulations.

To effectively transition from controlled laboratory experiments to real operational contexts, it is crucial to first understand the different modalities through which adversarial manipulations can occur. This distinction forms the foundation for designing realistic evaluation scenarios and selecting appropriate countermeasures.

5.1 Digital vs Physical Attacks

A key distinction in the study of adversarial attacks is between digital attacks and physical attacks. Digital attacks operate directly on the digital representation of an input, such as a raw image file, before it is fed into the model. These manipulations can be subtle, often imperceptible to humans, yet capable of causing severe misclassifications. Physical attacks, on the other hand, are designed to persist and remain effective when the manipulated object is captured through a sensor (e.g., a camera) in the physical world. Examples include adversarial stickers applied to stop signs or patterns printed on clothing to evade pedestrian detection. Both categories are important: digital attacks often represent the first step in identifying vulnerabilities, while physical attacks translate these weaknesses into tangible, real-world threats.

Two scenarios illustrate the criticality of this problem in the cited domains. The first is face recognition, which is widely deployed in security, authentication, and surveillance systems. In the digital setting, adversarial perturbations to facial images can lead to identity misclassification or evasion of recognition entirely, raising serious concerns for access control and personal privacy. Digital attacks in this space have been demonstrated to succeed even against models trained on large-scale, high-accuracy datasets, showing that accuracy alone is not a guarantee of robustness.

The second domain involves object detection in autonomous driving, where models must correctly identify and classify vehicles, pedestrians, and traffic-related objects. Here, we consider a dataset of car images in which adversarial manipulations simulate physical attacks, such as altered vehicle colors, wrap patterns, or localized texture perturbations, which could cause a model to misidentify a vehicle's type or fail to detect it entirely. These simulated physical attacks in a controlled, digital environment allow systematic testing of robustness while approximating the challenges of real-world adversarial conditions.

The significance of these scenarios lies in their high stakes: failures in facial recognition can compromise security and privacy, while failures in vehicle detection can threaten human safety on the road. Studying these applications not only validates the practical relevance of theoretical attack methods but also informs the development of countermeasures that can generalize beyond lab conditions. In the broader context of AI deploy-

ment, understanding and mitigating adversarial vulnerabilities in such critical domains is a foundational requirement for building systems that are both effective and secure in the environments they are meant to serve.

5.2 Face Recognition

In many publications, adversarial attacks on face recognition are demonstrated purely in academic contexts, without concrete, real-world threat scenarios. Inspired by the methodology of Physical Penetration Testing in cybersecurity, we constructed a plausible operational scenario.

Ethical Problem

Face recognition is a high-stakes application where adversarial attacks pose significant risks. In security or access-control systems, carefully crafted perturbations can allow an unauthorized individual to be classified as someone else, leading to breaches of privacy, identity theft, and physical security compromises.

Consider a security or surveillance system protecting a sensitive facility. An attacker obtains a publicly available photo of an authorized person (Individual B) and applies a targeted adversarial attack to another person's photo (Individual A). The perturbation is designed so that, to a human observer, the modified image still clearly appears to be Individual A, with no visible artifacts, as shown in Figure 8. However, when processed by the recognition backbone, the embedding is shifted into the region corresponding to Individual B, as visible in Figure 11. In this way, the attacker could present the altered image to gain access to restricted areas without authorization. Because the perturbation is visually imperceptible, the attack would be "invisible" to human guards and undetectable without specialized countermeasures. These risks are further amplified when models are trained on biased datasets, which can already yield unfair outcomes for certain demographic groups. The addition of adversarial examples not only reduces technical reliability but also undermines fairness, safety, accountability, and public trust in AI-driven identity verification systems.

Figure 6: Original query image.

Figure 7: Adversarial query image.

Figure 8: Original vs Adversarial query.

Figure 9: Original query's neighbors.

Figure 10: Adv query's neighbors.

Figure 11: Original vs Adversarial neighbors.

This scenario highlights that the most dangerous attacks may evade both machine and human scrutiny, and how important it is to study valuable defences.

Experimental Setup

For this scenario, we employ the publicly available Face Recognition Dataset from Kaggle [8], which contains approximately 5,000 labeled images of 31 distinct identities and is well-suited for classification tasks.

Our approach follows a two-stage training strategy. In the first stage, the backbone network is trained as a standard classifier on the dataset, while in the second stage it is fine-tuned for face recognition using metric learning, specifically with ArcFace loss. Although the final objective relies on metric learning, the initial classification phase is essential to obtain a baseline classifier for generating adversarial samples and to later study how robustness transfers across tasks.

To evaluate model performance, we adopt a structured query-based evaluation:

1. Randomly select 100 query images from the dataset.
2. For each query, compute its 30 nearest neighbors in the CNN feature space.
3. Measure classification accuracy over these neighbors and average across the 100 queries to obtain a representative score.

This evaluation is carried out on both clean and adversarially perturbed datasets, considering multiple attack strategies:

- **PGD (Projected Gradient Descent)**: PGD severely disrupts the model's internal feature space, producing predictions unrelated to the original query identity and inconsistent across neighbors. This indicates that the attack destroys the global representation structure rather than merely causing targeted misclassifications.
- **Carlini & Wagner L2**: In contrast, the C&W attack shifts the query representation toward a consistent but incorrect cluster of identities. This often results in the query being associated with the majority of samples from a single (wrong) identity, suggesting a more targeted manipulation of the embedding space compared to PGD.

By quantifying accuracy degradation in this controlled setup, we clearly demonstrate both the feasibility and severity of adversarial attacks on real-world face recognition systems.

Countermeasures

After identifying adversarial training as the most effective defense strategy against the considered vulnerabilities, we adopt this approach by incorporating PGD-generated adversarial examples directly into the training process. This integration enables the model to learn parameters that are inherently more resilient to adversarial attacks. In our experiments, the adversarially trained model demonstrates consistently high inference accuracy across all evaluated scenarios: the clean (unaltered) dataset, the dataset perturbed with PGD, and the dataset attacked with the C&W method. These results indicate that adversarial training can produce a model with strong, generalized robustness, capable of resisting some of the most common and powerful adversarial attacks.

Limitations

Despite providing valuable insight, this scenario has certain constraints. The dataset, although significant for our application, is still small compared to large-scale industrial systems, potentially limiting generalization. Furthermore, all attacks in our experiment are conducted in the digital domain, where the attacker has direct access to the image pixels. In the physical world, adversarial perturbations can be introduced via printed patterns, adversarial glasses, or altered clothing, which can be significantly harder to detect and defend against.

Experimental Results

The tests we conducted show that both PGD and C&W attacks are capable of severely compromising the model's ability to correctly predict identities. However, some important differences emerged:

- **Overall accuracy drop**: On 100 query images, PGD caused a much higher misclassification rate. Specifically, the accuracy dropped from 100% to 25.8%, while the C&W attack reduced accuracy to 59%. Although the latter is less severe, such a level is still far below what would be acceptable in a face recognition system.
- **Quality of the attack**: Although PGD achieves a larger drop in accuracy, the quality of the misclassifications makes C&W far more concerning in a real-world

scenario. In fact, PGD produces seemingly random misclassifications, causing the model to associate the query with identities that are both different from the query and from each other. By contrast, C&W is more subtle: it tends to make the model consistently misclassify the query as the same or very similar identities, effectively simulating an identity swap, as evident in Figure 14.

These findings highlight once again the gap between purely theoretical evaluations and practical cases. The real danger lies in real-world vulnerabilities: while random misclassifications already undermine system reliability, the most critical threat in face recognition is the possibility of a targeted and consistent identity swap, which has direct implications for security and privacy.

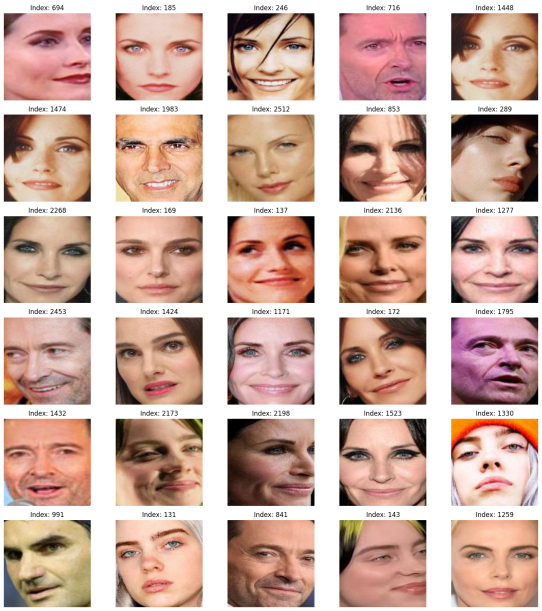

Figure 12: Query's neighbors with PGD attack.

Figure 13: Query's neighbors with C&W attack.

Figure 14: Comparison of neighbors retrieved after PGD vs C&W attacks.

Given these striking results, which highlighted the fragility of the baseline model, we next sought to improve robustness through adversarial training. As anticipated, we augmented the dataset with PGD-generated adversarial samples and retrained the model on the classification task. The model was then fine-tuned on the metric learning task, this time using the clean dataset. This strategy led to a substantial performance boost: accuracy on PGD-attacked samples increased from 0.25 to 0.76, while accuracy on C&W-attacked samples rose from 0.59 to 0.87. Crucially, this improvement was achieved without compromising performance on clean data, where accuracy remained virtually unchanged at 0.99.

These results underscore not only the robustness gained through adversarial training on a regular classification scenario, but also its transferability across tasks and loss functions, yielding remarkably strong performance.

Summarizing the results, we obtain the following comparison:

	Original Model Accuracy	Adversarially Trained Model Accuracy
Original Dataset	1.00	0.99
Adversarial (PGD)	0.25	0.76
Adversarial (C&W)	0.59	0.87

Table 11: Comparison of classification accuracy between the original and adversarially trained model across different datasets.

5.3 Vehicle Recognition in Security, Surveillance, and Autonomous Driving

Similar to the previous real-world scenario, we draw inspiration from Physical Penetration Testing to design a plausible setup that translates the theoretically studied risks of adversarial attacks into practical use cases. In this case, our focus is placed on the task itself rather than on specific attack or defense strategies. In particular, we investigate object detection, a widely deployed framework in real-world applications, and assess its vulnerabilities. The study of countermeasures is left for future work, given the higher cost and complexity required to develop effective defenses for such models.

Ethical Problem

Vehicle recognition in sensitive domains such as security, surveillance, and autonomous driving is a critical application where adversarial attacks raise serious safety and ethical concerns. Malicious perturbations targeting car detection systems can lead to misclassifications or missed detections, resulting in accidents, endangering human lives, and undermining trust in autonomous vehicles, or alternatively enabling security breaches, unauthorized access, and other surveillance failures.

Although the dataset used in our experiments resembles images from the surveillance domain, these vulnerabilities can compromise not only security and surveillance systems monitoring controlled-access areas such as garages, tunnels, or border checkpoints but also autonomous driving safety.

In these contexts, an adversary could wrap a vehicle with patterns derived from physically realizable adversarial attacks (e.g., camouflage or texture perturbations), causing the detection model to either fail to recognize the vehicle or misclassify it entirely.

This would allow a car to pass through an autonomously monitored zone without triggering alarms, undermining the integrity of automated access control, or, in the case of autonomous driving, cause potentially dangerous failures.

Such attacks could be carried out using adhesive patches, partial wraps, or printed covers, and require no electronic intrusion into the surveillance infrastructure.

This threat is particularly severe for fully automated surveillance or autonomous driving systems; however, even in hybrid settings where human supervision is partially present, the risk remains significant.

Below, a correctly detected instance is shown in Figure 15, while in Figure 16, the effects of adversarial patches causes a drastic drop in confidence or class misclassification,

and more often make the model totally miss any kind of detection, which is the most dangerous scenario.

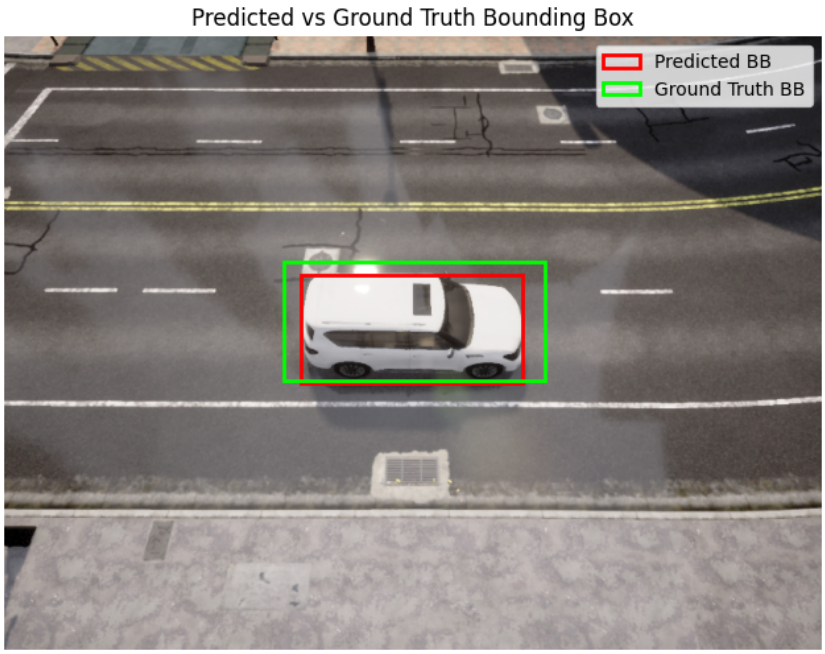

Figure 15: Original dataset car detection (red) and ground truth (green).

Figure 16: Adversarial dataset car detection (red) and ground truth (green) on different samples.

Experimental Setup

In this scenario, we use the publicly available adversarially augmented autonomous driving object detection dataset introduced by Yu et al. [15], which contains annotated images

with multiple objects under diverse adversarial perturbations. We focus specifically on the detection of cars, of which there are about 7000, out of which we sample 700 of them, testing a baseline object detector (using the Faster R-CNN architecture with a ResNet-50 backbone pretrained on the COCO Dataset) on this dataset.

Unlike manual attack generation, the dataset itself is pre-augmented with adversarial perturbations representing several attack types described in the original paper, simulating physical attacks, but still digitally, due to the lack of real physical datasets deployable. This leaves a further gap to be filled by creating real physical samples to test the algorithm more accurately, but still gives a reliable benchmark. In particular, the following attacks have been used:

- **FCA (Feature Collision Attack)**: Perturbs input images so that their feature embeddings collide with those of other objects, making distinct inputs map to similar internal representations and thereby confusing the detector.
- **DTA (Distributed Targeted Attack)**: Introduces small, spatially distributed perturbations over the object surface to force detectors into consistently misclassifying or missing the target, while remaining less perceptible than localized patch attacks.
- **ACTIVE (Adversarial Camouflage for Object Detection)**: Designs adversarial camouflage textures that, when overlaid on objects, make them inconspicuous to detectors. Unlike pixel-level noise, these perturbations appear as realistic textures or patterns.
- **3D2Fool**: A 3D adversarial attack that manipulates either the geometry or the texture of 3D objects to cause detection failures across multiple viewpoints, exploiting the fact that detectors must be robust to different viewing angles.
- **POOPatch (Patch-based Object Occlusion Attack)**: Places localized adversarial patches on objects (or in the scene) to occlude or distort discriminative regions, leading the detector to miss the object entirely or assign it an incorrect label.
- **RPAU (Robust Physical Adversarial Unrestricted Attack)**: Produces unrestricted and physically realizable perturbations (not limited to small ℓ_p-norms) that remain effective under diverse real-world conditions such as varying distances, lighting, and viewpoints.
- **CAMOU (Camouflage Attack)**: Generates camouflage textures that blend objects into their environment, reducing their visibility to detectors by lowering confidence scores and increasing the likelihood of missed detections in naturalistic settings.

For the inference phase, we use a threshold on the IoU equal to 0.5, as often chosen in the literature, to consider a bounding box as correctly matching a ground truth one.

To evaluate the attack's performance, we use the metrics explained in Section 4.5.

To evaluate model robustness, we measure standard object detection metrics (mean Average Recall (mAR), accuracy, and missed detections) on both clean and adversarially augmented subsets of the dataset, quantifying detection performance degradation induced by the adversarial perturbations.

Limitations

Although the dataset provides a realistic benchmark with diverse adversarial augmentations, it is limited to simulated perturbations and specific attack types as defined by Yu et al. [15]. Real-world adversarial attacks in autonomous driving may involve additional challenges, such as dynamic lighting, weather variations, and adversarial objects not covered by the dataset. Future work should focus on bridging this gap by incorporating physically realizable attacks and extending defenses to more varied real-world conditions.

Experimental Results

The experiments on the clean and adversarial dataset brought the results reported in Table 12 and Table 13.

	Missed Detection (out of 700 samples)	mAR	Model Accuracy
Clean Dataset	56	0.75	0.77

Table 12: Metrics on clean dataset on object detection

	Missed Detection (out of 700 samples)	mAR	ASR	Model Accuracy
3d2fool	67	0.59	0.21	0.62
Active	311	0.21	0.71	0.25
Appa	84	0.55	0.26	0.57
Camou	164	0.43	0.42	0.44
Dta	170	0.38	0.48	0.40
Fca	178	0.46	0.38	0.48
Poopatch	227	0.36	0.52	0.38
Random	107	0.49	0.35	0.50
Rpau	190	0.42	0.43	0.44

Table 13: Metrics for different attacks in the object detection framework on the baseline model.

The results clearly show that adversarial perturbations can substantially undermine the performance of the object detection model. On clean data, the detector achieves strong accuracy (0.77) and recall (0.75), with relatively few missed detections. However, when attacks are applied, both accuracy and mAR drop consistently, and the number of images with no detections rises significantly. Among the tested methods, the *Active* attack emerges as particularly effective, causing the steepest decline in performance with accuracy falling to 0.25 and over 44% of images left undetected. Similarly, patch-based methods such as *Poopatch* and *Dta* also achieve high attack success rates (above 0.48), revealing their strong disruptive potential. These findings underline that certain attack strategies, especially active and patch-based ones, are able to compromise detection robustness to a severe extent, raising important concerns for the deployment of such systems in real-world scenarios.

Some visualization example are reported in Figure 17.

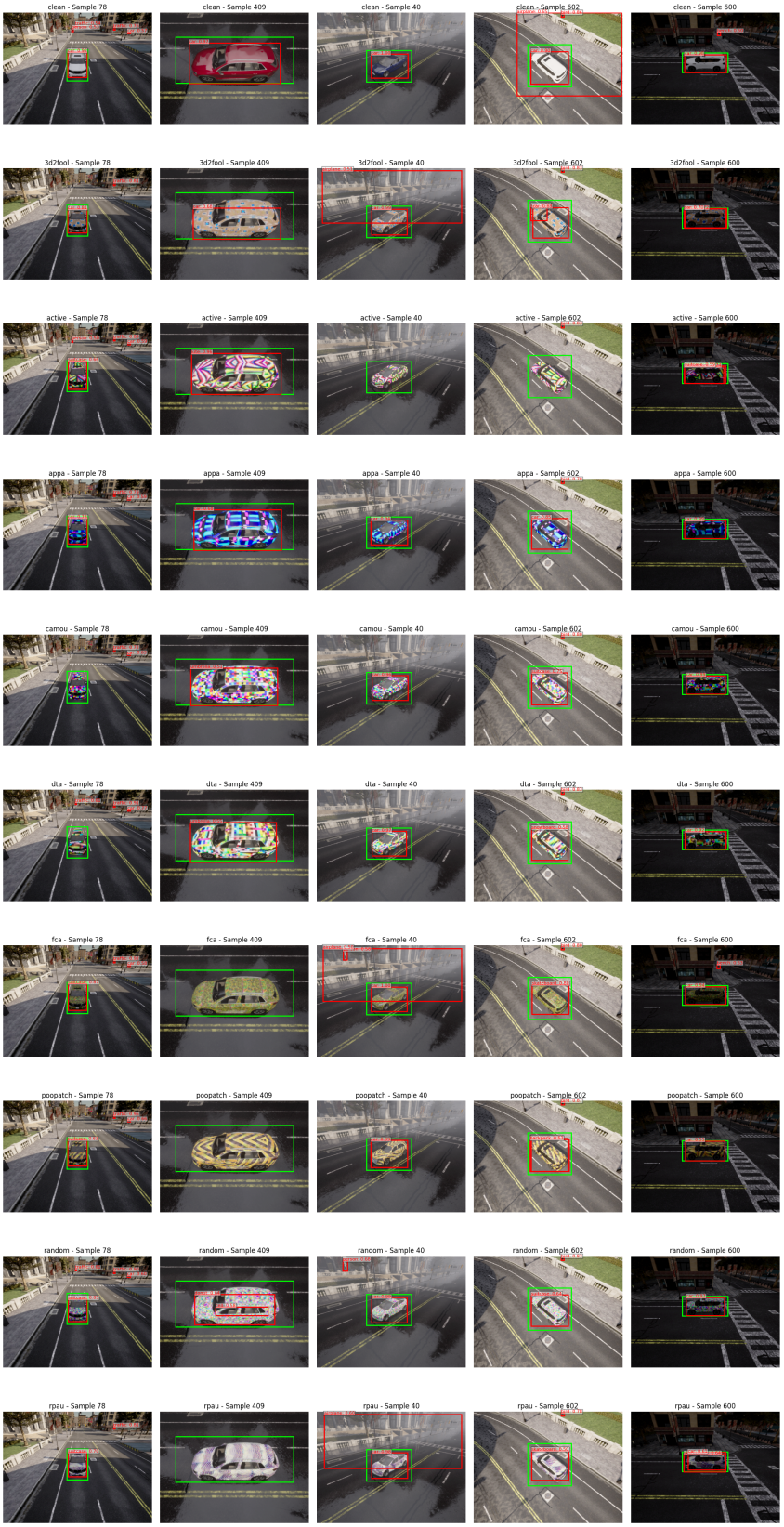

Figure 17: Visualization of different attacks' inference, detected bb (red) and ground truth (green).

6 Contribution

The project spanned roughly two months and was structured into two main phases: an initial study, where we evaluated the feasibility of the experiments and defined the workflow to follow, and a subsequent practical phase, during which the code was developed to run tests and a final report was prepared to summarize the work.

The first phase mainly focused on establishing shared guidelines and discussing the general direction of the project. The second phase, instead, involved a clearer division of tasks: Alessandro and Alex concentrated on code implementation and running experiments, while Mattia focused more on writing the report and conducting the discussion. Despite this division of responsibilities, all members remained actively involved, consistently updated each other, and contributed to the overall development of the project.

7 Conclusions

This work has addressed both the theoretical underpinnings and the practical exploitation of adversarial attacks in image-based AI systems. Beyond evaluating attack success rates and testing defense strategies, we dedicated particular attention to constructing and evaluating realistic scenarios that mirror the methodology of security penetration testing, looking for real-world applications of theoretically formulated studies. This approach bridges the gap between academic proof-of-concept demonstrations and concrete operational threats, which is one of the main challenges in the safety field.

From an ethical perspective, our findings underscore the responsibility of AI practitioners to anticipate malicious uses of their systems, even when such uses are not explicitly demonstrated in the research literature. We showed how attacks originally conceived in a purely digital context can be translated into physical, real-world threats, whether enabling an unauthorized person to pass a face recognition gate or allowing an unrecognized vehicle to enter a controlled area. By explicitly framing these plausible scenarios, we aim to foster awareness that robustness and security are not merely technical challenges, but essential requirements for trustworthy AI deployment.

Future work should extend this methodology to larger-scale datasets, physically realized adversarial patterns, and integrated defense pipelines combining detection, robust training, and context-aware access policies.

Overall, we are satisfied to have achieved our goal of not only assessing the theoretical challenges of adversarial attacks, but also grounding these insights in real-world scenarios, analyzing their potential dangers, and outlining possible paths toward effective solutions.

References

[1] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. *arXiv preprint arXiv:1608.04644*, 2017.

[2] Jianbo Chen, Michael I Jordan, and Martin J Wainwright. Hopskipjumpattack: A query-efficient decision-based attack. *arXiv preprint arXiv:1904.02144*, 2020.

[3] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with autoattack. *arXiv preprint arXiv:2003.01690*, 2020.

[4] Amin Ghiasi, Ali Shafahi, and Tom Goldstein. Breaking certified defenses: Semantic adversarial examples with spoofed robustness certificates. *arXiv preprint arXiv:2003.08937*, 2020.

[5] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

[6] Aleksander Madry et al. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[7] 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)*, pages 2574–2582, 2016.

[8] Vasuki Patel. Face recognition dataset, 2021. Kaggle dataset.

[9] Sara Sabour, Yanshu Cao, Fartash Faghri, and David J Fleet. Adversarial manipulation of deep representations. *arXiv preprint arXiv:1511.05122*, 2015.

[10] Jiawei Su, Danilo Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 2019.

[11] Christian Szegedy et al. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

[12] Tsui-Wei Weng et al. Evaluating the robustness of neural networks: An extreme value theory approach. *arXiv preprint arXiv:1801.10578*, 2018.

[13] Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*, 2020.

[14] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2018.

[15] Youran Zhang et al. Adversarial attack and defense for deep learning-based object detection: A survey. *arXiv preprint arXiv:2408.09181*, 2024.