Adversarial Lasers on the Road: A Threat to Pedestrian Detection in Autonomous Vehicle Systems

Submitted in partial fulfilment of the requirements

of the degree of

Bachelor of Technology (B.Tech)

by

Aman Seervi (21CSB0B01)

Aman Gupta (21CSB0B02)

Ankit Kumar (21CSB0B04)

Supervisor:

Dr. Venkanna Udutalapally
Associate Professor

Department of Computer Science and Engineering

NIT Warangal

2025

Acknowledgment

We would like to express our heartfelt gratitude to Dr. Venkanna Udutalapally, our project guide, for his valuable guidance, supervision, suggestions, and support throughout the semester in completing our project work. His encouragement during challenging times gave us the motivation to keep moving forward.

We would also like to thank the evaluation committee for their valuable feedback and for conducting a smooth and insightful project presentation.

Aman Seervi
21CSB0B01
B.Tech Final Year

Aman Gupta
21CSB0B02
B.Tech Final Year

Ankit Kumar
21CSB0B04
B.Tech Final year

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Aman Seervi	Aman Gupta	Ankit Kumar
21CSB0B01	21CSB0B02	21CSB0B04

Date:

Approval Sheet

This Project Work entitled **Adversarial Lasers on the Road: A Threat to Pedestrian Detection in Autonomous Vehicle Systems** by **Aman Seervi**, **Aman Gupta** and **Ankit Kumar** is approved for the degree of **Bachelor of Technology in Computer Science and Engineering.**

Examiners

_____

_____

_____

Supervisor(s)

_____

_____

_____

Head Of Department

_____

Date: _____ Place: _____

Certificate

This is to certify that the Project work entitled *Adversarial Lasers on the Road: A Threat to Pedestrian Detection in Autonomous Vehicle Systems* is a bonafide record of work carried out by *Mr Aman Seervi (21CSB0B01)*, *Mr Aman Gupta (21CSB0B02)* and *Mr Ankit Kumar (21CSB0B04)* submitted to the faculty of *Computer Science and Engineering*, in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in *Computer Science and Engineering* at National Institute of Technology, Warangal during the academic year 2024-25.

Dr R Padmavathy
Head of the Department
Department of Computer Science and Engineering
NIT Warangal

Dr. Venkanna Udutalapally
Associate Professor
Department of Computer Science and Engineering

Abstract

Autonomous vehicles have become more common in recent years. These rely on deep neural networks for tasks such as object detection, lane tracking, and pedestrian detection. While these models perform well in many cases, they remain vulnerable to adversarial attacks that can cause misclassification and lead to unsafe decisions. In this work, we propose a laser-based attack targeting night-time pedestrian detection systems. The attack works by projecting laser spots on pedestrians in a way that interferes with the model's ability to detect them. To determine effective spot placements, we use a greedy optimization approach on a reduced search space, which makes the process faster and suitable for real-time execution. The attack does not rely on any pre-positioned markers or prior knowledge of the scene.

We evaluate the attack on the Nano, Small, and Medium versions of the YOLOv8 object detection model. We measure its effectiveness using Attack Success Rate (ASR) and assess execution time to confirm real-time feasibility. Our method achieves an ASR of 82.7% with 5 laser spots and completes execution in under one second, showing that it can be carried out in realistic conditions. We also perform an ablation study to understand the effect of the laser wavelength and the number of restarts on attack performance. The aim of this work is not just to demonstrate a new attack, but also to contribute towards understanding these vulnerabilities so that future research can focus on designing more robust detection systems and developing effective defense mechanisms.

Contents

List of Figures

List of Tables

List of Algorithms

Abbreviations and Notations

Abbreviations

- AV – Autonomous Vehicle
- DNN - Deep Neural Network
- YOLO – You Only Look Once (object detection model)
- IoU – Intersection over Union
- ASR – Attack Success Rate
- mAP – Mean Average Precision
- ECDF – Empirical Cumulative Distribution Function

Notations and Symbols

- λ – Weighting factor in loss function; also denotes laser intensity (context-dependent)
- r – Radius of laser spot
- α – Intensity decay parameter
- $\mathscr{M}$ – Object detection model
- $\mathbf{I}$ – Input image
- $b_i = (x_i, y_i, w_i, h_i)$ – Bounding box coordinates and dimensions
- L_i – Class label for bounding box b_i
- $\mathscr{L}_{\text{loc}}$ – Localization loss (e.g., IoU or Smooth L1)
- $\mathscr{L}_{\text{cls}}$ – Classification loss (e.g., cross-entropy or focal loss)
- $\mathscr{D}$ – Dataset used for training
- δ – Adversarial perturbation
- ε – Maximum allowable perturbation norm
- $\|\cdot\|_p$ – Norm constraint (e.g., L_2, L_∞)

Chapter 1

Introduction

Autonomous vehicles (AVs) have the potential to revolutionize transportation by providing safer and more efficient travel compared to traditional human-driven vehicles. Critical tasks must be performed with high accuracy without human intervention for the successful operation of an AV. One such task is pedestrian detection, where the system must detect the presence of pedestrians in images or video streams and accurately determine their locations. AVs rely heavily on deep neural networks (DNNs) to detect traffic signs, pedestrians, lanes, etc. Although DNNs have achieved high accuracy in performing these tasks, they are vulnerable to adversarial attacks: subtle, carefully crafted perturbations that can deceive the model into making incorrect predictions. Such vulnerabilities seriously threaten the safety and reliability of autonomous systems, as even minor disruptions in pedestrian detection could lead to catastrophic outcomes, including accidents and loss of life. Recent research has demonstrated the feasibility of adversarial attacks in digital environments and in the physical world.

Currently, adversarial attacks can be categorized into digital and physical attacks. Most of the early work in this area was on digital attacks, in which an attacker is assumed to have the power to manipulate the input images at a pixel level. FGSM [1] and PGD [2] are typical digital attacks. Although highly effective, their practicality in real-world scenarios is limited because they rely heavily on directly altering the input images, which is not always feasible outside of digital settings. Physical attacks address this limitation by introducing perturbations directly into the physical environment before a camera captures the input. Earlier physical attacks [3]–[5] used stickers that could be attached to the target object to fool the model. Thys et al. [6] proposed a similar patch-based method specific to pedestrian detection, and Xu et al. [7] demonstrated how these patches could be printed directly on t-shirts, making the attack less noticeable and more practical in real-world scenarios.

Patch-based techniques are effective but have a key limitation: the stickers must be placed in advance and cannot be easily altered, making them static and inflexible. This also prevents spontaneous attacks on arbitrary objects or pedestrians, as the patch needs to be prepositioned. To address these challenges, researchers have introduced light-based attacks. Gnanasambandam

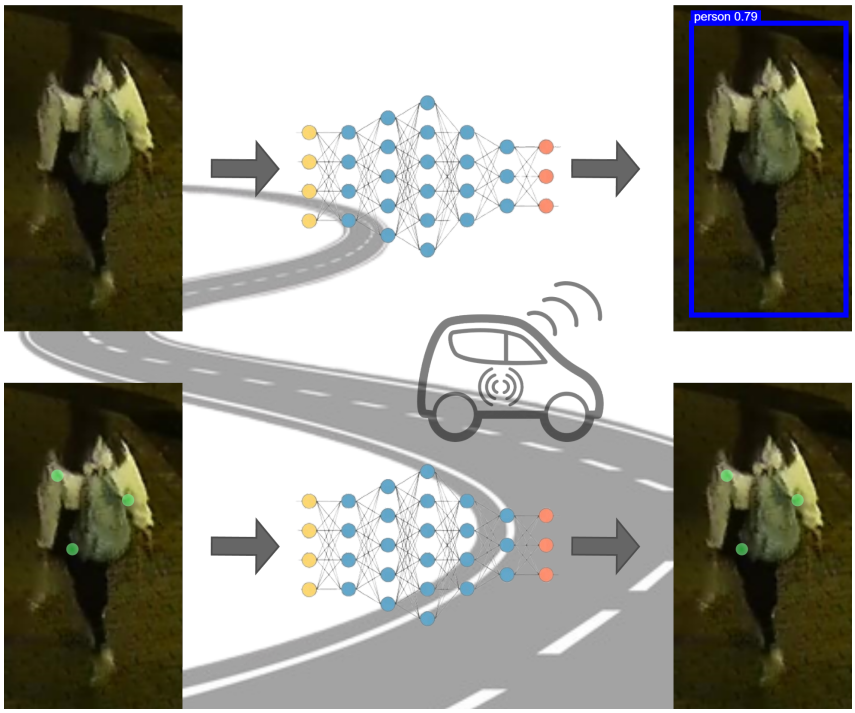


Figure 1.1: **Laser Spot Attack** : The detection model fails to detect the pedestrian when a few laser dots are strategically positioned.

et al. [8], and Lovisotto et al. [9] use projectors to cast carefully designed light patterns onto the target, allowing spontaneous attacks. In Duan et al. [10], the authors utilized a laser beam to carry out the attack, which is more practical as the laser is easy to handle and more stealthy. However, a key drawback in their approach was using a laser beam that sweeps across the image. In real-world settings, laser light may not scatter as much, and this sweeping motion can compromise the stealthiness of the attack. Moreover, while most of the previous work uses an optimization algorithm to tune attack parameters, they lack a detailed feasibility analysis regarding the time required to generate them. This raises concerns about their practicality and relevance in real-world scenarios.

In this paper, we propose a novel end-to-end adversarial attack using a laser against pedestrian detection at night. Our attack setup will essentially have an AV equipped with the ability to project multiple laser dots on objects in its vicinity and sufficient computational resources to perform calculations, run detection models, etc. The idea is that this vehicle can be dynamically deployed at any location to conduct an attack. The attack begins with the AV capturing the scene before it, running it through a detection model (e.g., YOLO) to identify pedestrians in the image. The system then digitally simulates the placement of laser spots on the detected pedestrians, optimizing their locations to maximize the attack's success rate. Once a suitable configuration is determined, the laser spots are projected in real-time to execute the attack. This causes other nearby AVs to misinterpret the altered input image, leading to incorrect decisions. Figure 1.1 shows such an attack example. Using laser spots gives us the following advantages:

1. Laser spots are small and thus more discreet compared to other light-based attacks such as projectors or sweeping laser beams.

2. The attack can be executed spontaneously on arbitrary pedestrians without prior preparation.

Given the need for real-time execution, achieving a high attack success rate within a minimal timeframe is critical. Prolonged optimization would compromise spontaneity, as the scene can change with vehicle movement. To address this, we propose an algorithm to change a continous search space into a discrete one and then use a greedy based optimization technique on this reduced search space. This allows us to carry the attack in real-time. We have conducted a feasibility analysis to evaluate the time required for a successful attack, aiming to minimize latency while maintaining an effective success rate. The contributions of this work can be summarized as follows.

1. We introduce a framework for light-based adversarial attacks specific to pedestrian detection at night. The method uses laser projections to place spots on pedestrians, enabling attacks to be executed without any prior setup. Unlike patch-based methods, it does not require physical attachment or prepositioning, allowing attacks on arbitrary pedestrians.

2. To enable real-time use, we develop an approach for quick identification of attack parameters. The search space is first converted from continuous to discrete, followed by a greedy-based optimization. This allows the algorithm to efficiently determine parameters like laser placement, achieving high success rates with minimal projections.

3. We conduct a feasibility analysis, demonstrating the practicality of our attack in real-world scenarios. An attack success rate of 82.74% is achieved against the YOLO nano model, with 50% of the attacks completing in under 0.22 s.

The paper is structured as follows: Chapter 2 covers related work, Chapter 3 provides necessary preliminaries, Chapter 4 explains our attack approach, Chapter 5 describes the experimental setup, and Chapter 6 presents the results.

Chapter 2

Review of Literature

2.1 Pedestrian Detection

Pedestrian detection involves identifying and precisely localizing pedestrians within an image, typically by drawing bounding boxes around them to indicate their position. This task is a fundamental aspect of autonomous vehicles, and accurate detection is crucial, as any failure or misclassification can lead to severe consequences, including potential collisions and loss of life.

Detection methods are broadly categorized into two-stage and single-stage approaches. Two-stage detectors, such as Fast R-CNN [11] and Faster R-CNN [12], first generate region proposals and then classify and refine them, leading to high accuracy but increased computational cost. In contrast, single-stage detectors, such as SSD [13] and YOLO [14], predict bounding boxes and class labels in a single pass, offering significantly faster inference times. The latest versions, such as YOLOv7 and YOLOv8, deliver real-time performance while maintaining accuracy comparable to or even better than two-stage models.

2.2 Adversarial Attacks

The adversarial attack involves deliberately crafting inputs to deceive machine learning models into making incorrect predictions. Szegedy et al.[15] first proposed the generation of adversarial samples by adding imperceptible perturbations to the inputs. Adversarial samples can be generated either digitally, by directly altering input data, or physically, by modifying real-world objects to mislead the model.

2.2.1 Digital Attacks

Digital adversarial attacks mislead deep learning models by adding subtle perturbations to the input, which humans cannot detect. Early works focused on limiting these perturbations within mathematical norms such as (l_2, l_0, l_∞) norm to ensure effective attacks while maintaining

stealth ([16],[17],[18]). These attacks are generally categorized into white-box and black-box settings. In white-box attacks, the attacker has full access to the target model, including its architecture and gradients, enabling direct optimization of adversarial examples [19]. Conversely, black-box attacks operate without internal knowledge of the model, relying instead on transferability or query-based approaches ([20], [21]). Gradient-based attacks such as FGSM [1] and iterative methods [18] refine perturbations, while momentum [22] and transformation techniques [23] enhance transferability.

Recent advances focus on improving stealth. Hosseini et al. [24] manipulated HSV color spaces, Laidlaw et al. [25] maximized perceptual similarity, and Kwon et al. [26] applied channel-specific perturbations, all to create adversarial examples that mimic natural variations and evade detection. Although digital adversarial attacks are effective, they assume that the attacker can directly modify the input image of the target model, which is not practical in real-world scenarios.

2.2.2 Physical Attacks

Physical adversarial attacks exploit real-world objects to deceive machine learning models. Kurakin et al. [19] first tested this by printing adversarial samples and recapturing them with cameras. Sharif et al. [27] used adversarial eyeglass frames to bypass facial recognition, while Brown et al. [3] introduced ”adversarial patches” - stickers that mislead classifiers. Duan et al. [4] and Eykholt et al. [5] applied similar sticker-based attacks to traffic signs, subtly altering them to evade detection. Xu et al. [7] designed T-shirts with adversarial patterns to fool pedestrian detectors, and Athalye et al. [28] 3D-printed adversarial objects resistant to viewpoint changes.

Despite their effectiveness, these methods have significant limitations. They require physical access, which is not always feasible, and demand extensive premeditation (e.g., printing, placement), making spontaneous execution impossible.

2.2.3 Light-Based Physical Attacks

Light-based attacks offer a stealthier alternative by manipulating illumination rather than modifying physical objects. Nguyen et al. [29] demonstrated this by projecting adversarial patterns onto faces, fooling facial recognition systems. Zhou et al.[30] proposed an infrared-based facial morphing technique to avoid detection. Lovisotto et al. [9] utilized projectors to cast patterns on traffic signs to carry out the attack. Gnanasambanda et al. [8] introduced OPAD, a portable projector-camera system for real-time attacks on 3D objects.

Although these methods reduce the reliance on physical modifications, many still require bulky, conspicuous equipment. Recent work prioritizes subtlety: Zhong et al. [31] manipulated natural shadows to create perturbations, while Duan et al. [10] proposed AdvLB, a laser-based

attack capable of introducing adversarial perturbations. AdvLB demonstrated real-world feasibility using a simple laser pointer. However, its broad beam makes it less covert, and the laser weakens when viewed from an angle, reducing its effectiveness over long distances. Hu et al. [32] introduced AdvLS, where the attack was conducted using laser spots, which are more subtle than beams. They projected 10-50 laser spots on an object to evade detection. However, they did not provide an estimate of the attack time, making it unclear if the attack is feasible in real time. Building on these advances, our work uses localized laser spots, a precise and discreet method that avoids broad laser sweeps' visibility and range issues. We specifically focus on targeting pedestrians and conduct a feasibility analysis to ensure that our attack can be carried out in real time. Table 2.1 presents a comparison between our work and previous studies.

Table 2.1: Comparison of related works

Related Works	**Pedestrian Specific**	**Physical /Light**	**Laser Based**	**Spontaneous Execution**	**Real-Time Feasibility**
Brown et al. [3]	X	Physical	X	X	X
Xu et al. [7]	✓	Physical	X	X	X
Lovisotto et al. [9]	X	Light	X	✓	✓
Gnanasambandam et al. [8]	X	Light	✓	✓	✓
Duan et al. [10]	X	Light	✓	✓	✓
Hu et al. [32]	X	Light	✓	✓	✓
Ours	✓	Light	✓	✓	✓

Chapter 3

Preliminaries

3.1 Pedestrian Detection

Pedestrian detection is the task of identifying and localizing pedestrians in an image. Formally, let $\mathbf{I} \in \mathbb{R}^{l \times w \times h}$ be an image with dimensions l (height), w (width), and h (number of channels, e.g., 3 for RGB images). The goal is to train a model $\mathscr{M}$ that maps an input image $\mathbf{I}$ to a set of bounding boxes and associated labels.

Let the output of the model be represented as:

$$\mathscr{M}(\mathbf{I}) = \{(b_1, L_1), (b_2, L_2), \ldots, (b_N, L_N)\} \tag{3.1}$$

where:

- $b_i = (x_i, y_i, w_i, h_i)$ denotes the i-th bounding box with coordinates (x_i, y_i) (top-left corner) and dimensions w_i (width) and h_i (height).
- $L_i \in \{0, 1\}$ is the label assigned to b_i, where $L_i = 1$ indicates the presence of a pedestrian and $L_i = 0$ indicates no pedestrian.

The model $\mathscr{M}$ is trained on a dataset $\mathscr{D} = \{(\mathbf{I}_j, \mathscr{G}_j)\}_{j=1}^{M}$, where each image $\mathbf{I}_j$ has a corresponding ground truth set:

$$\mathscr{G}_j = \{(b_1^*, L_1^*), (b_2^*, L_2^*), \ldots, (b_{N_j}^*, L_{N_j}^*)\} \tag{3.2}$$

with b_i^* and L_i^* denoting the ground truth bounding boxes and labels.

The objective of the model is to minimize the detection error across the dataset by optimizing the following loss function:

$$\mathscr{L}(\mathscr{M}, \mathscr{D}) = \sum_{j=1}^{M} \left(\sum_{i=1}^{N_j} \mathscr{L}_{\text{loc}}(b_i, b_i^*) + \lambda \sum_{i=1}^{N_j} \mathscr{L}_{\text{cls}}(L_i, L_i^*) \right) \tag{3.3}$$

Where:

- $\mathscr{L}_{\text{loc}}(b_i, b_i^*)$ is the localization loss, typically defined as Smooth L1 loss or IoU-based loss to measure the error in bounding box regression.
- $\mathscr{L}_{\text{cls}}(L_i, L_i^*)$ is the classification loss, commonly cross-entropy loss or focal loss, ensuring the correct classification of pedestrians.
- λ is a weighting factor that balances localization and classification losses.

Thus, the training task is:

$$\mathscr{M}^* = \arg\min_{\mathscr{M}} \mathscr{L}(\mathscr{M}, \mathscr{D}) \tag{3.4}$$

such that the model accurately detects pedestrians while minimizing false positives and false negatives.

3.2 Adversarial Attack

An adversarial attack aims to manipulate the predictions of a model by adding small and carefully crafted perturbations to the input data. Formally, consider an image $\mathbf{I} \in \mathbb{R}^{l \times w \times h}$ with a true label y. Let $\mathscr{M}$ be a model that maps inputs to predicted labels:

$$\mathscr{M} : \mathbb{R}^{l \times w \times h} \to \{0, 1\} \tag{3.5}$$

The adversarial example $\mathbf{I}_{\text{adv}}$ is generated by adding a perturbation δ to the original image $\mathbf{I}$:

$$\mathbf{I}_{\text{adv}} = \mathbf{I} + \delta \tag{3.6}$$

The attack aims to satisfy two conditions:

1. **Misclassification:** The model's prediction changes such that:

$$\mathscr{M}(\mathbf{I}_{\text{adv}}) \neq \mathscr{M}(\mathbf{I}) \tag{3.7}$$

2. **Imperceptibility:** The perturbation is small enough to be visually undetectable by humans. A constraint commonly enforces this:

$$\|\delta\|_p \leq \varepsilon \tag{3.8}$$

where $\|\cdot\|_p$ is a norm (e.g., L_2 or L_∞ norm) and ε is a small positive value.

The attacker's objective is to find δ that minimizes classification accuracy while satisfying the imperceptibility condition.

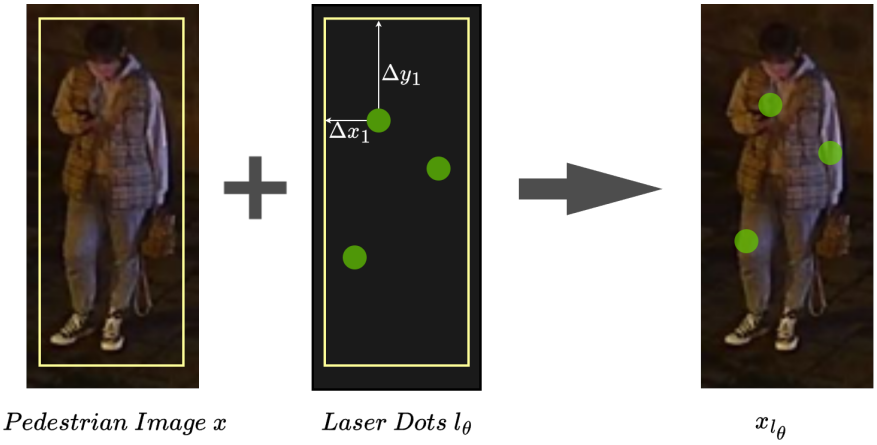


Figure 3.1: **Example attack :** The goal is to find suitable offset parameters Δx_i and Δy_i. Laser spot size not to scale.

In our context, as shown in Figure 3.1, the objective of the proposed attack is to determine the offset parameters θ such as Δx_i and Δy_i for each of the n points. These offsets represent the horizontal and vertical distances from the top-left corner of the bounding box. The goal is to get a modified image x_{l_θ} so that the model does not detect the pedestrian.

Chapter 4

Approach

This chapter is organized as follows. We first describe the attack setup. Next, we model the laser spots and define the relevant parameters. Finally, the optimization algorithms used to perform the attack is discussed.

4.1 Attack Setup

Our attack setup involves a malicious AV equipped with a camera and sufficient computational resources to handle detection and optimization tasks. The AV captures the scene and uses a detection model, such as YOLO, to identify pedestrians in the environment. Once pedestrians are detected, the system digitally simulates the placement of laser spots on them and optimizes these positions to maximize the effectiveness of the attack. After determining the locations, the laser spots are projected, causing other AVs to misinterpret the image. This leads them to fail to detect the pedestrian and make potentially incorrect decisions. A visual representation of this flow can be found in Fig 4.1. This setup is highly flexible, as the AV can be deployed at various locations with minimal preparation. Furthermore, the fully automated process ensures quick execution and precise laser positioning.

4.2 Laser Spot Definition

In this work, a laser spot is defined by a set of key parameters that characterize its physical properties and its effect on the target. The parameters are as follows:

- **Wavelength** (λ): The wavelength λ determines the color of the laser point. Only wavelengths within the visible range (380 nm to 750 nm) are considered. We define a conversion function that converts λ to an RGB tuple[1].
- **Location**: The location of each spot is defined by its offsets from the top left corner of the bounding box. For the i-th spot, let $(\Delta x_i, \Delta y_i)$ denote the horizontal and vertical offsets.

[1] `https://gist.github.com/friendly/67a7df339aa999e2bcfcfec88311abfc`

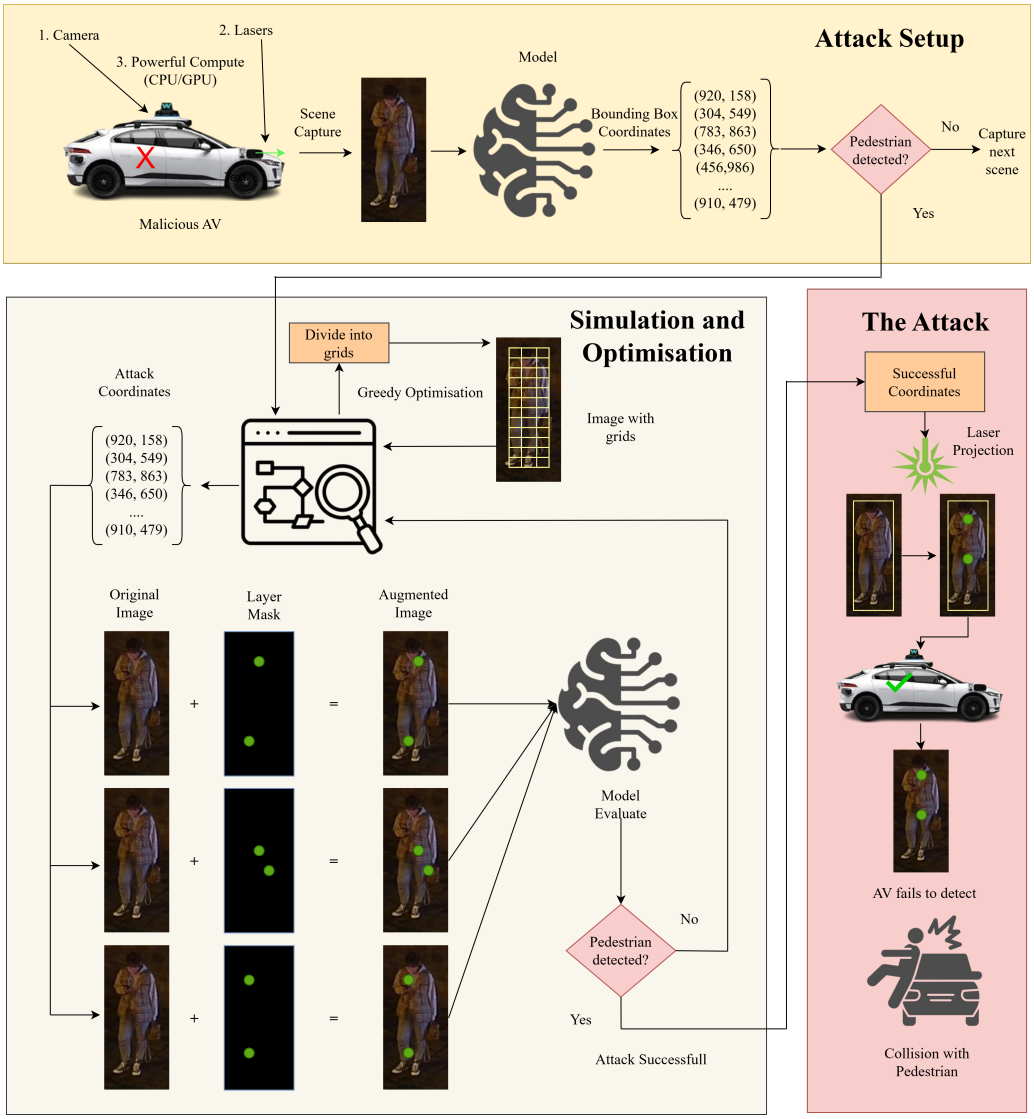


Figure 4.1: **Attack Flow :** The attack has three phases. First, the malicious AV captures the scene ahead. If a pedestrian is detected, the second phase computes the best laser spot placement. In the final phase, the AV projects the laser onto the pedestrian using its onboard laser, causing nearby AVs to fail in detection.

- **Radius** (r): The radius r defines the size of the laser point, representing the area it covers. We introduce a Gaussian blur to make the spot look more realistic and account for the natural scattering effect of laser light. The blurring operation is applied mathematically using a Gaussian filter with standard deviations $\sigma_X = \sigma_Y = r/3$.
- **Intensity** (α): The intensity α represents the brightness of the laser point. This parameter is based on the power of the laser device and can be adjusted according to the attack requirements.

Let the parameter vector for a single laser point be

$$\theta = (\lambda, \Delta x_i, \Delta y_i, r, \alpha). \tag{4.1}$$

A group of n laser spots is then represented as

$$G_\theta = \{\theta_1, \theta_2, \ldots, \theta_n\}. \tag{4.2}$$

A mask, denoted by l_θ, is generated from G_θ and fused with a clean image x to produce the adversarial image $x_{l\theta}$:

$$x_{l\theta} = x + l_\theta, \tag{4.3}$$

where the fusion is performed using a linear image fusion method.

4.3 Laser Spot Attack

Parameter Optimization

Given the set of parameters

$$\theta = (\lambda, \Delta x_i, \Delta y_i, r, \alpha) \tag{4.4}$$

We aim to optimize each laser spot so that the detection model fails to identify the pedestrian. Now, one would try to enforce strict constraints in many adversarial attack settings.

$$\varepsilon_{\min} \leq \theta \leq \varepsilon_{\max}, \tag{4.5}$$

Which helps to ensure that the modifications are minimal. However, in our case, these constraints are inherently satisfied by the laser's physical limitations and the small scale of the changes.

In this paper, we also focus on ensuring that the attack happens quickly, as delays may allow the scene to change, and the attack would then be futile. However, the offsets $(\Delta x_i, \Delta y_i)$ form a continuous search space, making the optimization process computationally demanding. To address this, the bounding box is first divided into discrete cells. This discretization converts the constant search space for the offsets into a finite set of candidate positions. The division is done so that the cells have a side length of $2r$, so that each cell can accommodate one laser spot. Now, the task reduces to selecting n cells from the discretized grid such that when projecting the laser at those parts, the detection model fails to detect the pedestrian. Fig ref shows a visual representation of the same.

Algorithm 1 shows a simple approach to partition the bounding box into discrete cells. The algorithm dynamically calculates the number of rows and columns based on the dimensions of the bounding box and the minimum cell size (lines 3-6). The center of each cell is treated as a valid candidate position, which is determined by iterating over all cells and calculating their centers (lines 8-12). The final output is a list of these candidate positions.

We now focus on the remaining parameters with the candidate offsets determined through the grid-based approach. The wavelength λ is fixed since the laser device emits light in a single

Algorithm 1 Grid-based Dot Position Generation

Require: Bounding box $B = [x_{\min}, y_{\min}, x_{\max}, y_{\max}]$, minimum cell dimensions $w_{\min}$ and $h_{\min}$
Ensure: Set of dot positions $\mathscr{D}$

$w \leftarrow x_{\max} - x_{\min}$
$h \leftarrow y_{\max} - y_{\min}$
$N_{\text{cols}} \leftarrow \max(3, \lfloor w/w_{\min} \rfloor)$
$N_{\text{rows}} \leftarrow \max(1, \lfloor h/h_{\min} \rfloor)$
$c_w \leftarrow w/N_{\text{cols}}$
$c_h \leftarrow h/N_{\text{rows}}$
$\mathscr{D} \leftarrow \emptyset$
for $i = 0$ to $N_{\text{rows}} - 1$ **do**
  for $j = 0$ to $N_{\text{cols}} - 1$ **do**
    $x \leftarrow x_{\min} + j \cdot c_w + c_w/2$
    $y \leftarrow y_{\min} + i \cdot c_h + c_h/2$
    $\mathscr{D} \leftarrow \mathscr{D} \cup \{(x, y)\}$
  end for
end for
return $\mathscr{D}$

color. Similarly, the radius r and intensity α are predetermined based on empirical observations. Therefore, the primary focus of the optimization is to select the optimal offsets for the n spots, ensuring minimal changes.

4.3.1 Greedy Algorithm for Cell Selection

The problem now concerns an optimization task: selecting n cells from the candidate positions. We use a simple, greedy approach to do this.

The proposed algorithm aims to find an optimal configuration of laser dots that, when added to an image, reduces the confidence score of a targeted pedestrian's bounding box B_p as detected by a YOLO model. The inputs to this algorithm include:

- **Image:** The original image on which the attack is performed.
- **YOLO Model (M):** The pre-trained model used for pedestrian detection.
- **Dot Positions ($\mathscr{D}$):** The set of all possible positions where laser dots can be placed.
- **Bounding Boxes (B):** The initial list of bounding boxes detected by M before the attack (there may be multiple pedestrians in the image)
- **Pedestrian Box (B_p):** The bounding box corresponding to the pedestrian being targeted.

The algorithm follows a greedy optimization approach, which works as follows: A random configuration of dots (θ) is generated from $\mathscr{D}$. The key step is the **Evaluate** function, which simulates the addition of these dots to the image and then passes the modified image through the

YOLO model M. The confidence score of pedestrians previously represented by B_p is calculated and recorded. If the YOLO model fails to detect this pedestrian now (i.e., the score is below a threshold), the attack is considered successful, and the configuration is immediately returned (Lines 2–4).

Algorithm 2 Greedy Optimization for Laser Spot Placement

Require: Image x, YOLO model M, candidate dot positions $\mathscr{D}$, bounding boxes B, current pedestrian box B_p, parameters: number of dots n, iterations I, restarts R, and max moves per dot K

Ensure: Best configuration θ^* and corresponding score s^*

1: $s^* \leftarrow \infty$, $\theta^* \leftarrow \emptyset$
2: **for** $r = 1$ to R **do**
3: Initialize $\theta \leftarrow$ random sample of n positions from $\mathscr{D}$
4: $s \leftarrow$ Evaluate(x, M, θ, B_p)
5: **for** $i = 1$ to I **while improvement do**
6: improvement $\leftarrow$ false
7: **for** $j = 1$ to n **do**
8: Shuffle $\mathscr{D}$ and select candidate moves $C \subset \mathscr{D}$ of size K
9: **for** each $c \in C$ **do**
10: $\theta' \leftarrow \theta$ with the j-th dot replaced by c
11: $s' \leftarrow$ Evaluate(x, M, θ', B_p)
12: **if** $s' < s$ **then**
13: $\theta \leftarrow \theta'$, $s \leftarrow s'$, improvement $\leftarrow$ true
14: **end if**
15: **end for**
16: **end for**
17: **end for**
18: **if** $s < s^*$ **then**
19: $\theta^* \leftarrow \theta$, $s^* \leftarrow s$
20: **end if**
21: **end for**
22: **return** θ^*, s^*

The algorithm then iteratively refines the dot configuration by adjusting one dot at a time while keeping others fixed, allowing it to focus on the most impactful adjustments at each step. This localized approach makes the algorithm "greedy" in nature, where decisions are made incrementally to achieve immediate improvement. During each *iteration*, the algorithm evaluates several new configurations by modifying the position of a single dot at a time and selecting the position that provides the best reduction confidence score in B_p's (lines 5-15). This

process continues until no further improvement is detected or the maximum number of iterations is reached. Multiple *restarts* are performed to avoid local minima by randomly initializing new configurations.

4.3.2 Tracking the Correct Bounding Box

It is worth mentioning that an implementation-level nuance occurs when applying this algorithm. When laser dots are digitally added to an image for attack simulation, the changes may affect the detection of other pedestrians in the vicinity and the target pedestrian. This leads to a situation where the confidence scores of multiple bounding boxes are updated. As the algorithm requires reporting the confidence score of the target after each iteration, it becomes challenging to identify which bounding box corresponds to the original target.

Algorithm 3 Calculate Intersection over Union (IoU)

Require: Bounding boxes: box1 $= (x_{\min 1}, y_{\min 1}, x_{\max 1}, y_{\max 1})$ and box2 $= (x_{\min 2}, y_{\min 2}, x_{\max 2}, y_{\max 2})$
Ensure: IoU value
$x^I_{\min} \leftarrow \max(x_{\min 1}, x_{\min 2})$
$y^I_{\min} \leftarrow \max(y_{\min 1}, y_{\min 2})$
$x^I_{\max} \leftarrow \min(x_{\max 1}, x_{\max 2})$
$y^I_{\max} \leftarrow \min(y_{\max 1}, y_{\max 2})$
$A_I \leftarrow \max(0, x^I_{\max} - x^I_{\min}) \times \max(0, y^I_{\max} - y^I_{\min})$
$A_1 \leftarrow (x_{\max 1} - x_{\min 1}) \times (y_{\max 1} - y_{\min 1})$
$A_2 \leftarrow (x_{\max 2} - x_{\min 2}) \times (y_{\max 2} - y_{\min 2})$
$A_U \leftarrow A_1 + A_2 - A_I$
return $\frac{A_I}{A_U}$ **if** $A_U > 0$, **else** 0

Algorithm 4 Find Relevant Bounding Box

Require: The set of bounding boxes $\mathscr{B}$ from the current iteration and the previous bounding box B_p to be tracked
Ensure: The bounding box B^* with the highest IoU with B_p
best_iou $\leftarrow 0$, $B^* \leftarrow \emptyset$
for each bounding box $B \in \mathscr{B}$ **do**
  iou $\leftarrow$ `calculate_iou`(B_p, B)
  if *iou* $>$ *best_iou* **then**
    best_iou $\leftarrow$ *iou*, $B^* \leftarrow B$
  end if
end for
return B^*

We use a function known as `find_relevant_bbox` to track the target pedestrian during the attack simulation consistently. It identifies the most relevant bounding box by calculating the Intersection over Union (IoU) metric to measure the overlap between bounding boxes. Mathe-

matically, IoU is defined as

$$\text{IoU} = \frac{A_I}{A_U}, \tag{4.6}$$

where A_I is the area of intersection and A_U is the area of union between two boxes (see Algorithm 3). This metric provides a quantitative measure of similarity between the bounding box of the previous iteration B_p and each candidate bounding box in the current iteration. The function `find_relevant_bbox` iterates over the candidate boxes, computes the IoU with each one (Lines 1–8 in Algorithm 4), and selects the one with the highest IoU. This selected bounding box is then treated as the updated target B_p for subsequent iterations. This approach makes sure we correctly track the target pedestrian by picking the bounding box that most closely matches the box from the previous iteration.

4.4 Projecting the Laser Spots

Once a successful attack configuration is found, the autonomous vehicle uses its onboard laser to project the calculated pattern onto the pedestrian. This results in nearby vehicles failing to detect the pedestrian, completing the attack process. The exact mechanism of the laser projection system—such as hardware design, alignment, and real-time control—is not discussed in this work and is considered out of scope. We assume that the AV has a laser system capable of projecting at the required coordinates with reasonable accuracy and speed.

Chapter 5

Experimental Setup

5.1 Models & Dataset

We use the YOLOv8 model for pedestrian detection, which offers multiple versions of varying sizes: nano, small, medium, large, and extra-large. As the model size increases, the mAP (mean Average Precision) improves, but the improvement diminishes progressively. The increase in model size also results in slower inference time. For reference, YOLOv8 Nano has an mAP of 37.3 on COCO, while the Medium model reaches 50.2—a 12.9-point gain. However, the Large and Extra Large models offer only slight improvements at 52.9 and 53.9, while requiring significantly more time, with GPU inference times of 2.39 ms and 3.53 ms, respectively. The nano and small models are fast enough for real-time applications, such as video processing, with GPU inference times under 1.2 ms. Hence, in this paper, we conduct the experiments on the nano, small, and medium versions of YOLOv8 to balance performance and inference speed. A detailed comparison of the model variants is shown in Table 5.1.

Table 5.1: YOLOv8 Model Performance Metrics[1]

Model	**Size**	**mAP@50-95**	**CPU (ms)**	**GPU (ms)**	**Params (M)**	**FLOPs (B)**
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

For our dataset, we use the LLVIP dataset [34], a visible-infrared paired dataset containing 30,976 images, or 15,488 image pairs. Most of these images are captured in dark scenes and include various road and traffic environments, providing diverse pedestrian appearances across

[1] Source: Ultralytics documentation [33].

different backgrounds and configurations. From this dataset, we select 15 distinct locations and sample 30 images from each location, creating a subset of 450 images for our experiments. This selection provides sufficient diversity while keeping the dataset manageable for our purposes.

5.2 Evaluation Metric

To evaluate the effectiveness of our laser spot adversarial attack, we use the metric known as *Attack Success Rate (ASR)*.

The ASR measures how often the attack successfully affects the detection of pedestrians. It is calculated as the ratio of the number of pedestrians on which the attack was successful to the total number of pedestrians the attack was attempted on.

Mathematically, this is expressed as:

$$\text{ASR} = \frac{N_{\text{success}}}{N_{\text{total}}} \tag{5.1}$$

Where:

- N_{success} is the number of pedestrians for which the attack was successful.
- N_{total} is the total number of pedestrians the attack was attempted on.

A higher ASR indicates a more effective attack.

Other than this, our feasibility study defines attack time as the duration taken by the algorithm to find a valid attack configuration, starting from dividing the image into a grid to identifying a suitable pattern. The time is reported in seconds.

5.3 Implementation Details

5.3.1 Parameter Values

For our experiments, the laser wavelength is set to 532 nm, which is a common choice as it corresponds to the standard green laser that is most readily available commercially. The spot radius r is fixed at 5 pixels, providing a balance between being small enough and sufficiently broad when projected on distant objects. A Gaussian blur is applied to simulate the natural diffusion of laser light, resulting in a more realistic appearance.For image fusion, the laser mask is blended with the original image using a weighted sum. This weight effectively controls the intensity of the laser spots. We choose the weight as 0.7, making them visible yet subtle in the final output.

The greedy optimization is configured with 10 iterations per restart, 3 restarts, and a maximum of 5 candidate moves per dot. These values were selected considering the time required for each optimization cycle. For the number of laser spots (n), experiments were conducted with

values ranging from 2 to 5. We limit the number of spots to 5, as using more than this makes the attack more noticeable. Overall, these values try to keep the search process reasonably fast while making sure the attack stays subtle.

5.3.2 Hardware

For our experiments, we used a device with an AMD Ryzen 7 (4000 series) CPU and an NVIDIA GTX 1650 GPU. To make use of the GPU's capabilities, the five candidate configurations per dot (i.e., max moves per dot) are simulated in parallel and processed simultaneously. This approach speeds up the evaluation process.

Chapter 6

Evaluation

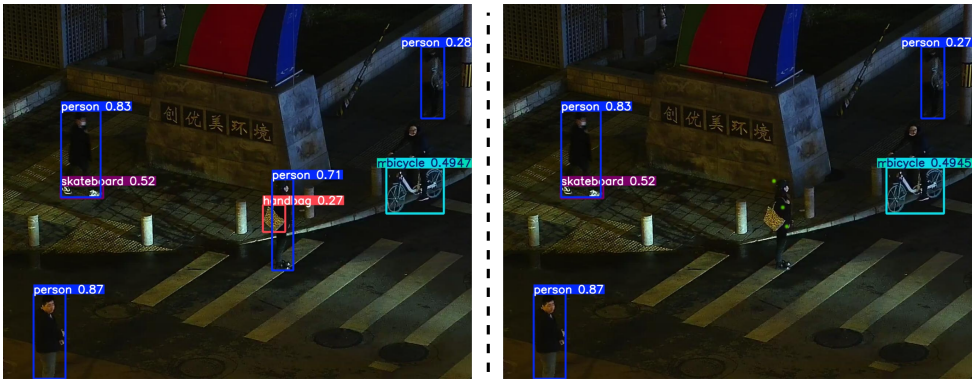
6.1 Attack Results

We now present the attack results. As mentioned earlier, we perform the attack on the Nano, Small, and Medium YOLOv8 models, varying the number of spots from 2 to 5. Table 6.1 shows the ASR for each combination. The highest ASR of 82.74% is achieved when attacking the Nano model with 5 dots. We observe that as the number of dots increases, the ASR also increases. This makes sense since more changes in the input create more chances for the model to make an error. Additionally, moving from the Nano to the Medium model, the attack becomes less effective. This is likely because larger models have more parameters, making them more robust. However, our method still achieves a 71.78% ASR against the Medium model, showing that the attack remains effective even as the model size increases. Figure 6.1 shows examples of the attack in action. Initially, the model detects the pedestrians, but fails to do so after projecting laser.

Table 6.1: Attack Results

Model	2 dots	3 dots	4 dots	5 dots
Nano	59.90%	69.80%	77.79%	82.74%
Small	52.43%	61.47%	79.38%	72.99%
Medium	48.34%	59.57%	65.14%	71.78%

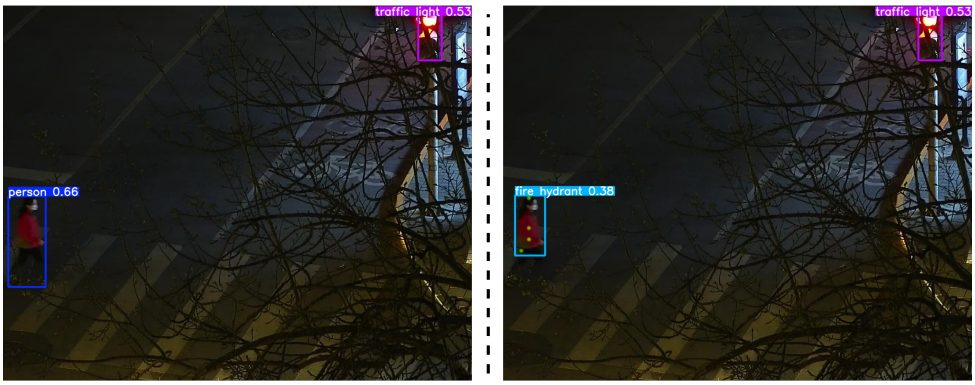
Table 6.2 compares our results with those of AdvLB [10] and AdvLS [32]. While AdvLB achieved 95.1% accuracy in a digital setting, their attack had no time constraints. In contrast, our method is designed specifically for pedestrian attacks and ensures efficiency by operating within a limited time. Our attack is also both harder to detect and easier to carry out than theirs.



(A)

(B)

(C)



(D)

Figure 6.1: Attack examples showing model output before (left) and after (right) the attack. In A and C, three strategically placed laser spots prevent the model from detecting the pedestrian. In B, the same effect occurs with four spots. In D, the pedestrian is misclassified as a fire hydrant after the attack.

Table 6.2: Attack Comparison with previous works

Approach	ASR	Efficiency Considerations
AdvLB. [10]	95.1%	No
AdvLS [32]	75.8%	No
Ours	82.74%	Yes

6.2 Feasibility Study

We now analyze the time required to successfully complete the attack. To better understand the distribution of attack completion times, we present quantile heatmaps in Figures 6.2 and 6.3 for the YOLO-Nano and YOLO-Small models, respectively. These heatmaps provide a visual representation of how the attack times vary with the number of laser dots used. In each heatmap, the cells represent the time (in seconds) by which a certain percentage of successful attacks were completed. Specifically, the heatmaps show the completion times for 50%, 75%, 90%, and 95% of successful attacks.

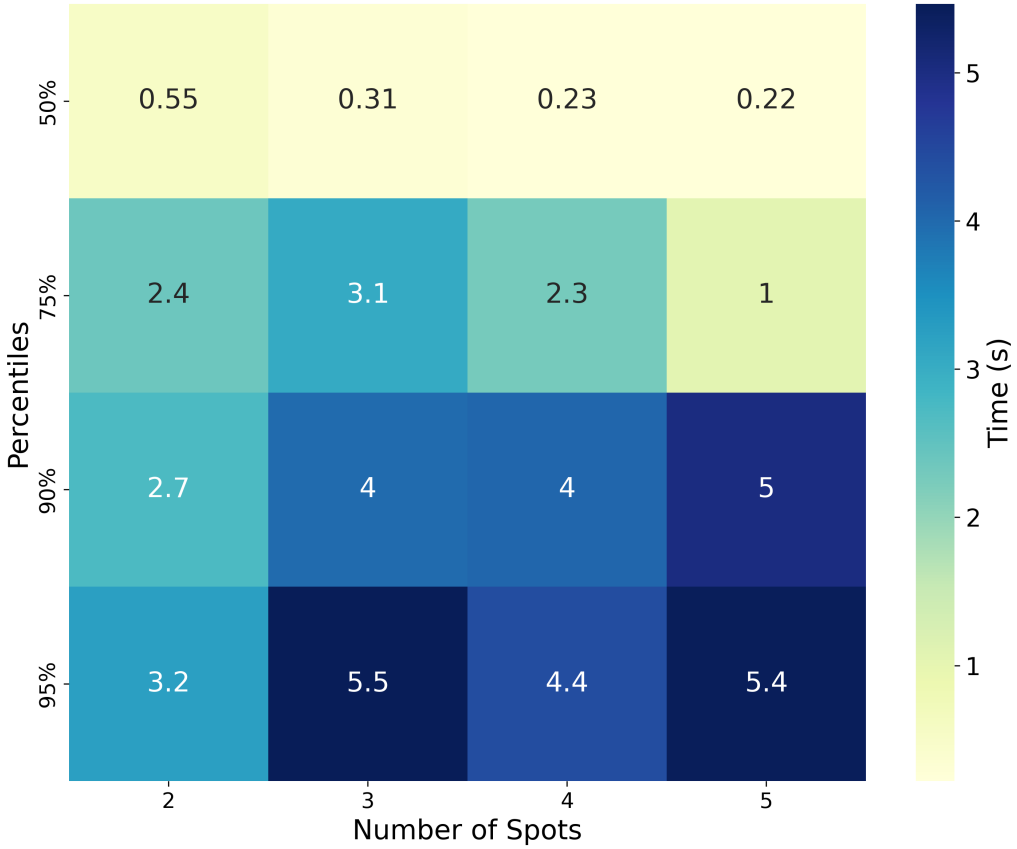


Figure 6.2: Quantile Heatmap - YOLO Nano

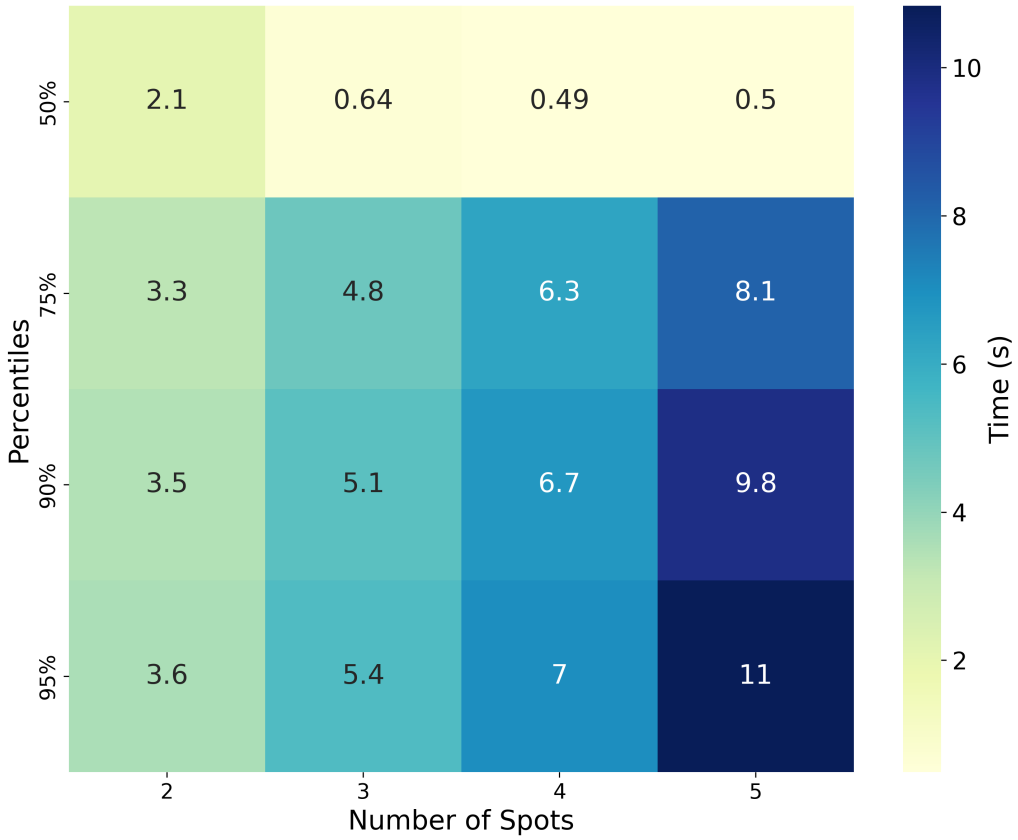


Figure 6.3: Quantile Heatmap - YOLO Small

With the Nano model and 5 dots, 50% of attacks succeed in under 0.22 s, and 75% succeed by 1.04 s. With the Small model (5 dots), 50% of attacks complete in 0.5 s, and for the Medium model (5 dots), that rises to 0.84 s. This shows that the attack is fast and can be practical for real-time scenarios.

We also see two main trends:

1. **Within each model size**, increasing the number of laser dots reduces the time needed for 50% of the attacks to succeed. This occurs because more dots make it easier to fool the model.

2. **Across different model sizes**, the median attack time becomes longer as the model size increases. This is mainly because larger models take more time to infer each image, which slows down the attack loop.

Figures 6.4, 6.5, 6.6, and 6.7 show the Empirical Cumulative Distribution Function (ECDF) plots for the YOLOv8 Medium model with 2, 3, 4, and 5 dots, respectively. Unlike the quantile heatmap, which shows specific percentiles, the ECDF provides a continuous view of how attack times are distributed.

We use these plots to highlight a pattern where the curve rises sharply at the start, then stays flat for some time before rising again. This means that some attacks succeed very quickly, while

others take more time to complete. The steep initial rise indicates that in some cases, the model is fooled early in the attack. The flat region shows that in other cases, the attack takes longer to succeed. One possible reason for this pattern is variation in the model's response to different dot configurations or image features. Some setups may trigger misclassification early, while others need more steps or frames to have an effect. Nonetheless, even with the Medium model when 5 dots are used, 50% of attacks succeed within 0.84 s, showing that the attack can still be effective within a short time. The general trends discussed can be seen here too.

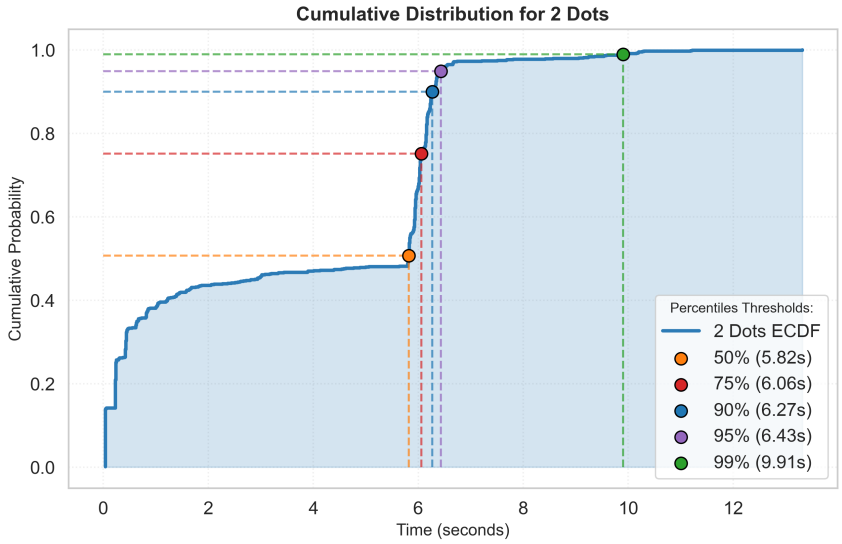


Figure 6.4: ECDF plot of attack times for YOLO Medium with 2 dots.

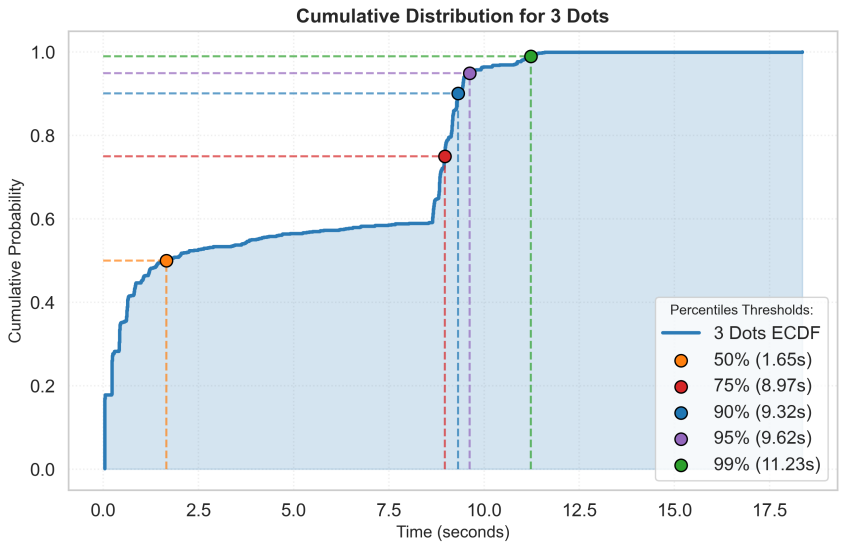


Figure 6.5: ECDF plot of attack times for YOLO Medium with 3 dots.

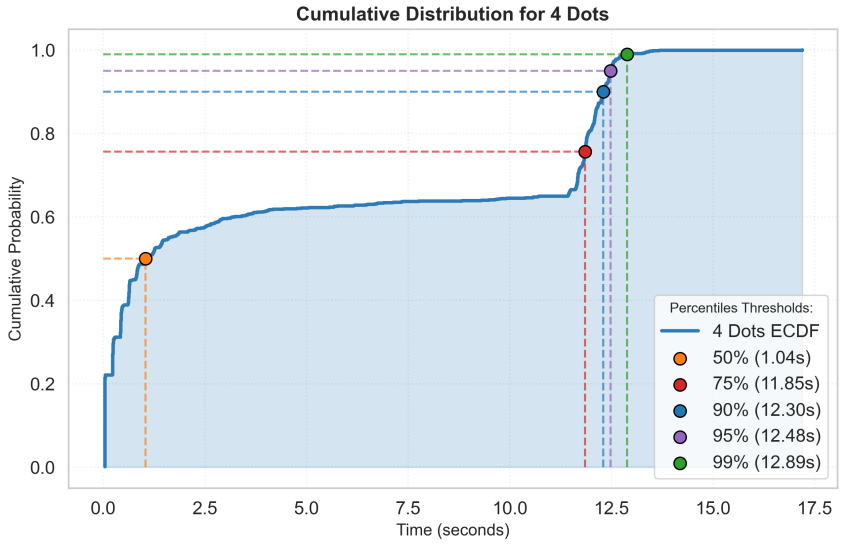


Figure 6.6: ECDF plot of attack times for YOLO Medium with 4 dots.

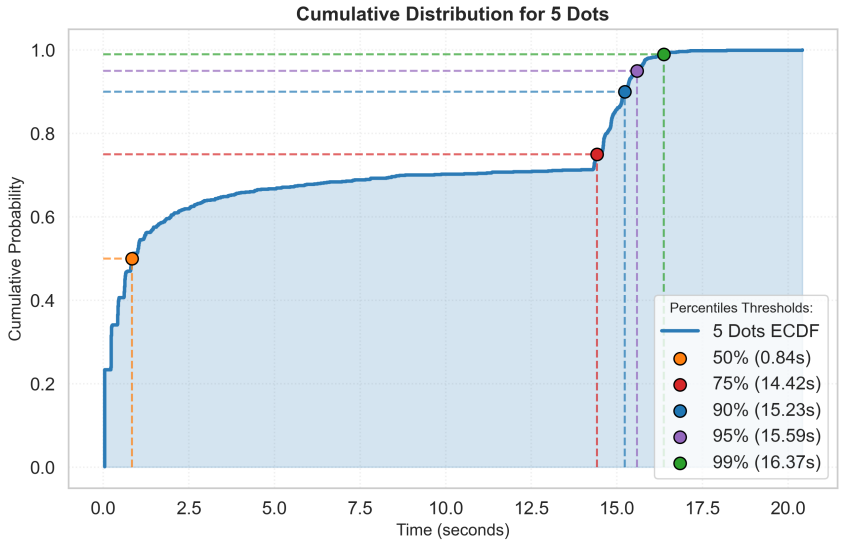


Figure 6.7: ECDF plot of attack times for YOLO Medium with 5 dots.

6.3 Ablation Study

In this section, we conduct a series of experiments to analyze how our attack is affected by different parameters. Each experiment helps us understand the role these factors play in the attack's effectiveness. Specifically, we examine:

1. Wavelength
2. Restarts

6.3.1 Wavelength

We analyze how the laser's wavelength λ affects the adversarial impact of the attack. To test this, we randomly place laser spots on each pedestrian and see if it is successfully evades detection. While this approach does not optimize for attack success, it provides a fair basis for comparing different wavelengths. We use the same sampled dataset as before, and vary the number of dots from 2 to 5 against the nano model. The process is repeated 5 times for each combination of wavelength, model size, and dot count and the average result is taken. Starting from 380 nm, we test wavelengths in 100 nm increments (up to 680 nm). The wavelength 532 nm is also included since it was used in all previous experiments. experiments.

Table 6.3: Performance Across Different Wavelengths

Wavelength	2 dots	3 dots	4 dots	5 dots
380 nm	4.89%	5.17%	8.35%	9.01%
480 nm	7.99%	10.81%	14.82%	17.79%
532 nm	7.81%	11.06%	14.51%	16.92%
580 nm	9.13%	12.86%	16.75%	20.07%
680 nm	4.89%	6.37%	8.93%	11.67%

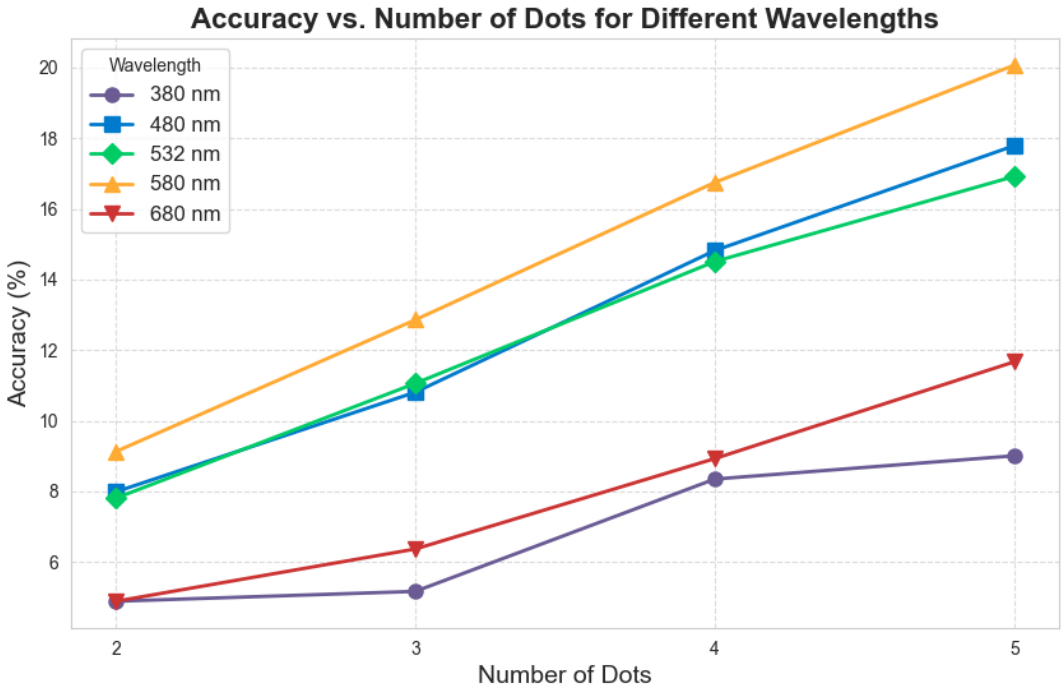


Figure 6.8: Ablation results on wavelength

Table 6.3 shows the performance across different wavelengths. The best results are observed at 580 nm, where the attack achieves the highest success rate (up to 20.07% with 5 dots). A

clear trend emerges, with performance increasing from 380 nm to 580 nm and then decreasing at 680 nm. The 532 nm wavelength, used in our earlier experiments, performs slightly below 580 nm but is still effective.

Figure 6.8 plots the corresponding results.

6.3.2 Restarts

To study the effect of restarts on the attack, we vary the number of restarts from 1 to 20. All experiments are performed on the same sampled dataset, using the YOLOv8-Nano model and 3 laser dots. Table 6.4 shows that increasing the number of restarts improves the Attack Success Rate (ASR) from 63.83% at 1 restart to 75.76% at 20 restarts. This indicates that repeating the attack from different initial conditions increases the chance of success. However, more restarts also require more computation time. So, depending on the application, a balance must be chosen between success rate and runtime.

Table 6.4: ASR vs. Number of Restarts

Restarts →	1	3	5	10	20
ASR →	63.83%	69.67%	71.19%	72.84%	75.76%

Chapter 7

Conclusion

In this paper, we presented a method for executing laser-based adversarial attacks against pedestrian detection. Our approach manipulates object detection by projecting multiple laser spots onto pedestrians, misleading the model into misclassifications. We formulated the attack as an optimization problem and implemented a greedy search algorithm to determine the best laser spot placements in real time. We conducted extensive experiments on three versions of the YOLOv8 model—Nano, Small, and Medium—evaluating the attack success rate (ASR) and execution time. Our results show that with just 5 laser spots, the ASR reached 82.7% on the Nano model, with half of the attacks completing in under 0.22s. The attack was found to be effective across different model sizes, although larger models exhibited slightly lower ASR values. Our approach remains feasible in real time, with execution times within practical limits. We also performed ablation studies on wavelength of laser and the number of restarts.

Future work should advance on two fronts. First, efforts should be directed toward developing defense mechanisms capable of detecting or mitigating such light-based adversarial attacks in real time. This includes sensor-level defenses, model improvements, and anomaly detection techniques.Second, the attack itself can be further improved by reducing execution time through better search algorithms, parallel computation, or deployment on high-end GPUs. Future work should also explore learning-based strategies, such as deep reinforcement learning or heuristics, to optimize laser spot placement. Additionally, testing in real-world conditions and evaluating more detection models can help assess robustness and generalizability.

References

[1] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[2] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *stat*, vol. 1050, no. 9, 2017.

[3] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer, "Adversarial patch," *arXiv preprint arXiv:1712.09665*, 2017.

[4] R. Duan, X. Ma, Y. Wang, J. Bailey, A. K. Qin, and Y. Yang, "Adversarial camouflage: Hiding physical-world attacks with natural styles," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 1000–1008.

[5] K. Eykholt, I. Evtimov, E. Fernandes, *et al.*, "Robust physical-world attacks on deep learning visual classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1625–1634.

[6] S. Thys, W. Van Ranst, and T. Goedemé, "Fooling automated surveillance cameras: Adversarial patches to attack person detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2019, pp. 0–0.

[7] K. Xu, G. Zhang, S. Liu, *et al.*, "Adversarial t-shirt! evading person detectors in a physical world," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, Springer, 2020, pp. 665–681.

[8] A. Gnanasambandam, A. M. Sherman, and S. H. Chan, "Optical adversarial attack," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 92–101.

[9] G. Lovisotto, H. Turner, I. Sluganovic, M. Strohmeier, and I. Martinovic, "{Slap}: Improving physical adversarial examples with {short-lived} adversarial perturbations," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 1865–1882.

[10] R. Duan, X. Mao, A. K. Qin, *et al.*, "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*, 2021, pp. 16 062–16 071.

[11] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[13] W. Liu, D. Anguelov, D. Erhan, *et al.*, "Ssd: Single shot multibox detector," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, Springer, 2016, pp. 21–37.

[14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[15] C. Szegedy, W. Zaremba, I. Sutskever, *et al.*, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[16] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proceedings of the 10th ACM workshop on artificial intelligence and security*, 2017, pp. 3–14.

[17] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 ieee symposium on security and privacy (sp)*, Ieee, 2017, pp. 39–57.

[18] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: A simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2574–2582.

[19] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Artificial intelligence safety and security*, Chapman and Hall/CRC, 2018, pp. 99–112.

[20] W. Brendel, J. Rauber, and M. Bethge, "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models," *arXiv preprint arXiv:1712.04248*, 2017.

[21] J. Chen, M. I. Jordan, and M. J. Wainwright, "Hopskipjumpattack: A query-efficient decision-based attack," in *2020 ieee symposium on security and privacy (sp)*, IEEE, 2020, pp. 1277–1294.

[22] Y. Dong, F. Liao, T. Pang, *et al.*, "Boosting adversarial attacks with momentum," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9185–9193.

[23] C. Xie, Z. Zhang, Y. Zhou, *et al.*, "Improving transferability of adversarial examples with input diversity," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2730–2739.

[24] H. Hosseini and R. Poovendran, "Semantic adversarial examples," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1614–1619.

[25] C. Laidlaw, S. Singla, and S. Feizi, "Perceptual adversarial robustness: Defense against unseen threat models," *arXiv preprint arXiv:2006.12655*, 2020.

[26] H. Kwon and S. Kim, "Dual-mode method for generating adversarial examples to attack deep neural networks," *IEEE Access*, 2023.

[27] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition," in *Proceedings of the 2016 acm sigsac conference on computer and communications security*, 2016, pp. 1528–1540.

[28] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," in *International conference on machine learning*, PMLR, 2018, pp. 284–293.

[29] D.-L. Nguyen, S. S. Arora, Y. Wu, and H. Yang, "Adversarial light projection attacks on face recognition systems: A feasibility study," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 814–815.

[30] Z. Zhou, D. Tang, X. Wang, W. Han, X. Liu, and K. Zhang, "Invisible mask: Practical attacks on face recognition with infrared," *arXiv preprint arXiv:1803.04683*, 2018.

[31] Y. Zhong, X. Liu, D. Zhai, J. Jiang, and X. Ji, "Shadows can be dangerous: Stealthy and effective physical-world adversarial attack by natural phenomenon," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 15 345–15 354.

[32] C. Hu, Y. Wang, K. Tiliwalidi, and W. Li, "Adversarial laser spot: Robust and covert physical-world attack to dnns," in *Asian conference on machine learning*, PMLR, 2023, pp. 483–498.

[33] Ultralytics, *Yolov8 documentation: Performance metrics*, `https://docs.ultralytics.com/models/yolov8/#performance-metrics`, Accessed: April 12, 2025, 2023.

[34] X. Jia, C. Zhu, M. Li, W. Tang, and W. Zhou, "Llvip: A visible-infrared paired dataset for low-light vision," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 3496–3504.

Aman thesis

ORIGINALITY REPORT

13%	9%	9%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.mdpi.com Internet Source	1%
2	export.arxiv.org Internet Source	1%
3	arxiv.org Internet Source	1%
4	Submitted to Cranfield University Student Paper	1%
5	dokumen.pub Internet Source	1%
6	Ruizhe Hu, Ting Rui, Yan Ouyang, Jinkang Wang, Qunyan Jiang, Yinan Du. "Light Attack: A Physical World Real-Time Attack Against Object Classifiers", IEEE Access, 2022 Publication	<1%
7	Moraffah, Raha. "Responsible Machine Learning: Security, Robustness, and Causality", Arizona State University, 2024 Publication	<1%
8	"Computer Vision – ECCV 2020", Springer Science and Business Media LLC, 2020 Publication	<1%
9	Amira Guesmi, Muhammad Abdullah Hanif, Muhammad Shafique. "AdvRain: Adversarial Raindrops to Attack Camera-Based Smart Vision Systems", Information, 2023 Publication	<1%

10	Submitted to University College London Student Paper	<1 %
11	Submitted to University of Wales, Lampeter Student Paper	<1 %
12	downloads.hindawi.com Internet Source	<1 %
13	researchbank.swinburne.edu.au Internet Source	<1 %
14	smartech.gatech.edu Internet Source	<1 %
15	www.citizen-systems.co.jp Internet Source	<1 %
16	Submitted to Macquarie University Student Paper	<1 %
17	helse-bergen.no Internet Source	<1 %
18	Jinlai Zhang, Lyujie Chen, Binbin Liu, Bo Ouyang, Qizhi Xie, Jihong Zhu, Weiming Li, Yanmei Meng. "3D Adversarial Attacks Beyond Point Cloud", Information Sciences, 2023 Publication	<1 %
19	Submitted to University of Surrey Student Paper	<1 %
20	Yuanwan Chen, Yalun Wu, Xiaoshu Cui, Qiong Li, Jiqiang Liu, Wenjia Niu. "Reflective Adversarial Attacks against Pedestrian Detection Systems for Vehicles at Night", Symmetry, 2024 Publication	<1 %
21	Zongqing Zhao, Shaojing Su, Junyu Wei, Xiaozhong Tong, Liushun Hu. "Improving Infrared Pedestrian Detection by Sharing Visible Light Domain Information with	<1 %

Enhanced UNet and YOLO Models", 2023 IEEE 16th International Conference on Electronic Measurement & Instruments (ICEMI), 2023
Publication

22	Submitted to Griffth University Student Paper	<1 %
23	Submitted to University of Huddersfield Student Paper	<1 %
24	ar5iv.labs.arxiv.org Internet Source	<1 %
25	www.ijac.net Internet Source	<1 %
26	Goodrum, Richard A.. "Algorithms and metrics for territorial design.", Proquest, 2014. Publication	<1 %
27	Yan Ding, Qingxin Cao, Bozhi Zhang, Peilin Li, Zhongjiao Shi. "Research on multi-view collaborative detection system for UAV swarms based on Pix2Pix framework and BAM attention mechanism", Defence Technology, 2024 Publication	<1 %
28	www.eaiib.agh.edu.pl Internet Source	<1 %
29	Amira Guesmi, Muhammad Abdullah Hanif, Bassem Ouni, Muhammad Shafique. "Physical Adversarial Attacks for Camera-Based Smart Systems: Current Trends, Categorization, Applications, Research Challenges, and Future Outlook", IEEE Access, 2023 Publication	<1 %
30	Liu, Yao. "Human-Centric Spatio-Temporal Modeling and Analysis for Multimedia Data.", University of New South Wales (Australia) Publication	<1 %

31	Mostofa, Moktari. "Generative Adversarial Network and its Application in Aerial Vehicle Detection and Biometric Identification System", West Virginia University, 2023 Publication	<1 %
32	ce-publications.et.tudelft.nl Internet Source	<1 %
33	digitalcommons.unl.edu Internet Source	<1 %
34	optimai.eu Internet Source	<1 %
35	www.cs.put.poznan.pl Internet Source	<1 %
36	www.math.helsinki.fi Internet Source	<1 %
37	www.slideshare.net Internet Source	<1 %
38	Hussain, Shehzeen Samarah. "Robust and Efficient Deep Learning for Multimedia Generation and Recognition", University of California, San Diego, 2023 Publication	<1 %
39	Wu, Han. "Practical Adversarial Attacks Against Deep Learning Models.", University of Exeter (United Kingdom), 2024 Publication	<1 %
40	papers.academic-conferences.org Internet Source	<1 %
41	"Applied Cryptography and Network Security Workshops", Springer Science and Business Media LLC, 2020 Publication	<1 %

42	"Computer Vision – ECCV 2018", Springer Science and Business Media LLC, 2018 Publication	<1 %
43	Jiawei Lian, Shaohui Mei, Shun Zhang, Mingyang Ma. "Benchmarking Adversarial Patch Against Aerial Detection", IEEE Transactions on Geoscience and Remote Sensing, 2022 Publication	<1 %
44	deepai.org Internet Source	<1 %
45	digitalassets.lib.berkeley.edu Internet Source	<1 %
46	ebin.pub Internet Source	<1 %
47	web.archive.org Internet Source	<1 %
48	www.coursehero.com Internet Source	<1 %
49	www.ecva.net Internet Source	<1 %
50	"Computer Vision – ECCV 2016", Springer Nature, 2016 Publication	<1 %
51	Chengyin Hu, Weiwen Shi, Ling Tian. "Adversarial color projection: A projector-based physical-world attack to DNNs", Image and Vision Computing, 2023 Publication	<1 %
52	Man, Yanmao. "Attacks and Defenses on Autonomous Vehicles: From Sensor Perception to Control Area Networks", The University of Arizona, 2022 Publication	<1 %