Disrupting Pedestrian Detection in AVs: A Study on Laser-Induced Adversarial Attacks

Final Year Project mid-review Presentation by :
Aman Seervi (21CSB0B01)
Aman Gupta (21CSB0B02)
Ankit Kumar (21CSB0B04)

Dept. of Computer Science & Engineering, NIT Warangal

Under the Guidance of: Prof. Venkanna U

Autonomous Vehicles Overview

- A vehicle capable of sensing its environment and operating without human input.
- **Key Components:**
  - **Sensors:** Lidar, Cameras, IR Cameras, GPS
  - **Detection & Decision Making:** AI models for object identification and decision accuracy
  - **Actuators:** Execute decisions for vehicle control
- AI must be highly accurate to prevent accidents, as errors could lead to severe consequences.

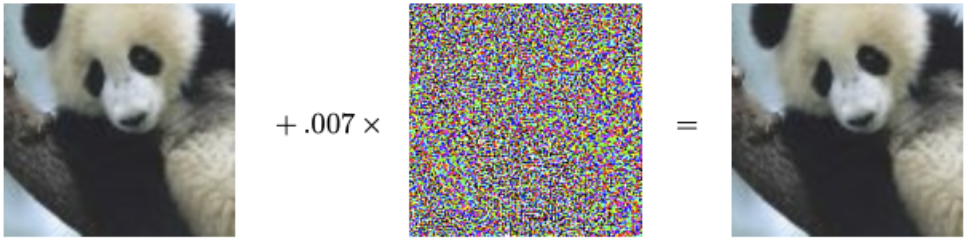
[6]

Adversarial Attack

- **Definition:** Intentional manipulation of inputs to deceive AI models into making incorrect predictions or decisions.
- **Exploitation:** Leverages vulnerabilities in the model's learning or decision-making process.
- **Types of Attacks:**
  - **White Box Attack**
  - **Black Box Attack**
  - **Physical Attack**

White Box Attack

- Full knowledge of the AI model's architecture and parameters.
- Example: Fast Gradient Sign Method (FGSM) [9]



Black Box Attack

- No or limited knowledge of the AI model.
- Probing the model to infer how to manipulate inputs.

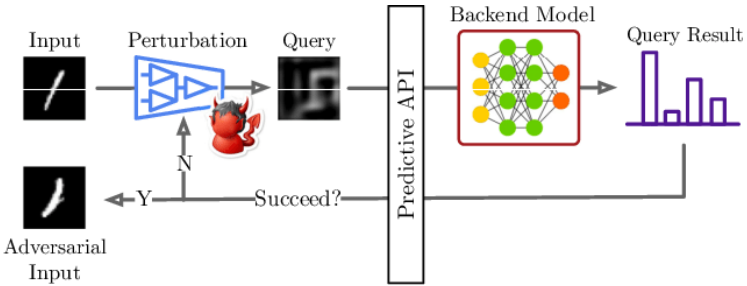


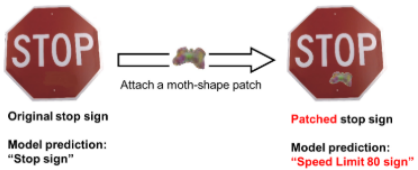
Figure: Illustration of a Black Box Attack [8]

Physical Attack

Definition: Physical attacks involve altering real-world objects or environments to deceive AI systems.

Key Characteristics:

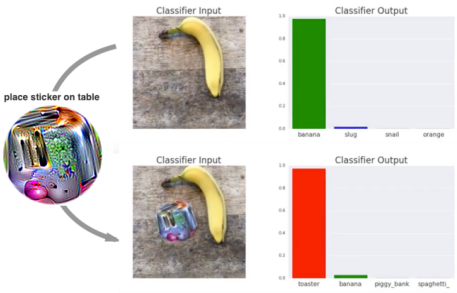
- Examples:
  - Sticking patches on traffic signs.
  - Creating 3D objects that are misidentified by AI systems.



- Challenges the robustness of AI models in physical environments.
- Must involve *subtle* changes — too noticeable changes defeat the attack's purpose.

Previous Work: Classic Attacks

Adversarial Patch [1]



Adversarial T-shirt [11]

Light Based Physical attacks in AVs

SLAP: Improving Physical Adversarial Examples with Short-Lived Adversarial Perturbations [7]

(a) Non adversarial scenario. (b) Adversarial projection.

Essentially projector based attack

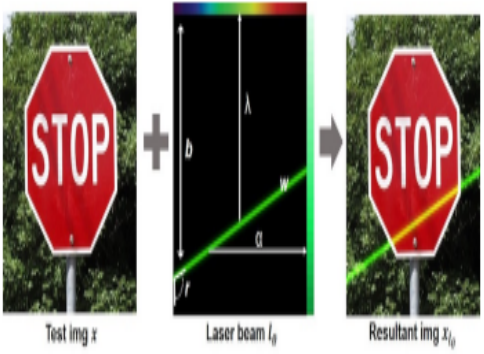
WIP: Adversarial Retroreflective Patches: A Novel Stealthy Attack on Traffic Sign Recognition at Night [10]

Attack using reflective patches

Continuation

Reflective Adversarial Attacks against Pedestrian Detection Systems for Vehicles at Night [2]

Adversarial Laser Beam: Effective Physical-World Attack to DNNs in a Blink [3]



Comparison of Adversarial Attacks

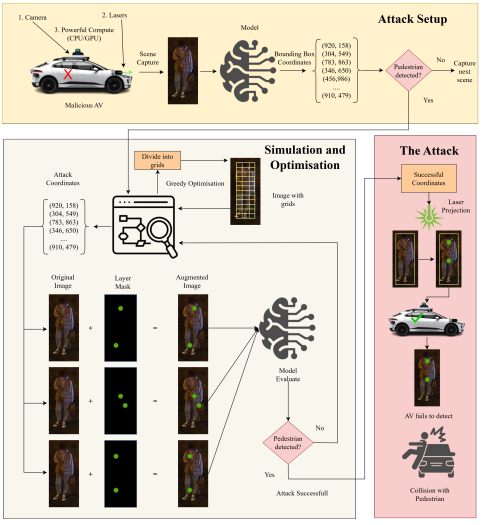
Method	Pedestrian-Specific	Physical/Light	Laser-Based	Spontaneous Execution	Real-Time Feasibility Analysis
Brown et al. [1]	✗	Physical	✗	✗	✗
Xu et al. [11]	✓	Physical	✗	✗	✗
Lovisotto et al. [7]	✗	Light	✗	✓	✗
Tsuruoka et al. [10]	✗	Light	✗	✗	✗
Chen et al. [2]	✓	Light	✗	✗	✗
Duan et al. [3]	✗	Light	✓	✗	✗
Ours	✓	Light	✓	✓	✓

Motivation

- **Criticality in AVs:** Adversarial attacks are critical in autonomous vehicles due to severe potential consequences.
- **Focus on Pedestrian Detection:** Targeted attacks on pedestrian detection systems in AVs due to the high risk involved.
- **Limitations of Existing Methods:** Reflective patches require pre-planning and aren't universally effective.
- **Shift to Light-Based Attacks:** Moved to light-based attacks for broader scope, but projector-based methods lack subtlety.
- **Laser-Based Attack Exploration:** Explored laser-based attacks due to challenges with projector methods.
- **Existing Laser Attack Issues:**
  - Visible beam in depictions vs. less pronounced effect in reality.
  - Attacker positioning in front makes it detectable and impractical.

Methodology

- An AV equipped with the ability to point laser dots on objects in front of it.
- It captures the scene and uses a model like YOLO to detect pedestrians.
- An algorithm determines optimal laser dot positions for manipulation.
- The AV projects laser dots, altering the scene perceived by the target AV, leading to misclassification.



Laser Spot Definition

- **Wavelength (λ)**: Determines color; visible range (380–750 nm). Mapped to RGB.
- **Location**: Offsets $(\Delta x_i, \Delta y_i)$ from top-left of bounding box.
- **Radius (r)**: Spot size; Gaussian blur with $\sigma = r/3$.
- **Intensity (α)**: Brightness based on laser power.
- Spot vector: $\theta = (\lambda, \Delta x_i, \Delta y_i, r, \alpha)$
- Group of spots: $G_\theta = \{\theta_1, \theta_2, \ldots, \theta_n\}$
- Image fusion: $x_{l_\theta} = x + l_\theta$

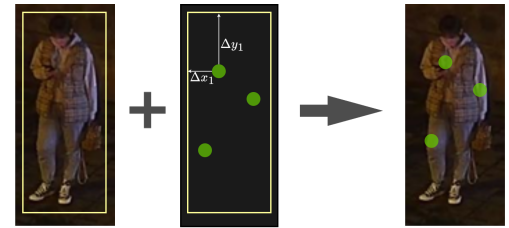


Figure: Attack Flow Diagram

Grid-Based Offset Optimization

- The attack must be fast — delays can make it ineffective.
- Continuous offset search is computationally expensive.
- To simplify, the bounding box is divided into discrete grid cells of size $2r \times 2r$.
- Each cell center becomes a valid candidate position for placing a laser spot.
- The goal is to select n such positions that cause pedestrian detection to fail.

Optimizing laser placement

- We adopt an optimization-based approach.
- A greedy algorithm iteratively refines dot placements to maximize attack success while remaining undetectable.
- Each step perturbs the positions slightly and checks if detection confidence decreases.
- If confidence drops, the change is accepted; otherwise, a new perturbation is tested.
- The process runs for multiple iterations to refine placements effectively.
- To avoid getting stuck in suboptimal configurations, multiple restarts are used.
- The best configuration is retained for the final attack execution.

Algorithm for greedy optimization

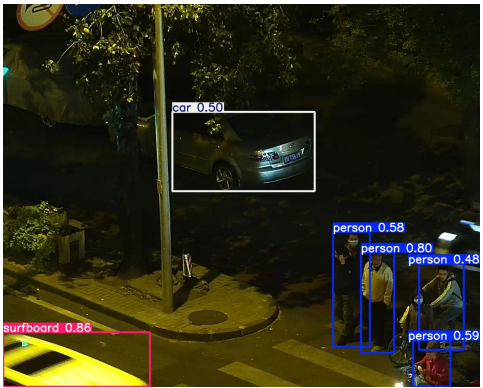
Input: Image x, YOLO model f, candidate positions P, n dots, max iterations T, restart attempts R
Output: Optimized dot placement θ^*

1. Initialize best configuration θ^*.
2. For $r = 1$ to R :
3. Randomly select n dot positions from P.
4. While improvement possible:
5. For each dot d :
6. Perturb d with new position.
7. Run YOLO model f, get confidence.
8. If confidence improves, update θ^*.
9. If attack successful, return θ^*.
10. Return best placement θ^*.

Tracking Bounding Box Across Iterations

Problem:

- YOLO gives multiple detections per frame.
- Dots may alter confidence values of nearby boxes
- Need to track the correct pedestrian box
- Simple confidence tracking fails—other boxes' scores change too
- Can't just track the highest-confidence box



Proposed Solution: IoU-based Box Association

Approach:

- Use **Intersection over Union (IoU)** to associate bounding boxes between iterations.
- Let B_{i-1} be the pedestrian's bounding box in iteration $(i-1)$ with confidence c_{i-1}.
- At iteration i, select the bounding box B_i that maximizes IoU with B_{i-1}.

$$B_i = \arg\max_{B \in \mathcal{B}_i} \text{IoU}(B, B_{i-1})$$

where $\mathcal{B}_i$ is the set of all bounding boxes in iteration i.

Experimental Setup - Part 1

- **Inference Model**: YOLOv8 nano,small and medium
- Fast, state-of-the-art object detection
- Used by Waymo for AV perception
- One-shot detection, up to 200 FPS on GPUs
- **Dataset**: LLVIP dataset [5]
- Nighttime urban road images with pedestrians
- 450 images sampled (30/scene, 15 locations)

Experimental Setup - Part 2

- **Attack Simulation**:
- Laser projection simulation with computer vision
- Laser spots: Gaussian blurring + weighted blending
- Process:
  - Pass images through YOLO to detect pedestrians
  - Algorithm decides laser direction
  - Place laser dots on detected locations
  - Re-feed images to model; attack successful if pedestrians undetected

- **Attack Success Measurement**: ASR = $\frac{P_{\text{before}} - P_{\text{after}}}{P_{\text{before}}}$
- P_{before}: pedestrians detected before attack
- P_{after}: pedestrians detected after attack
- Higher ASR = more effective attack

Attack Results

Table: Attack Results

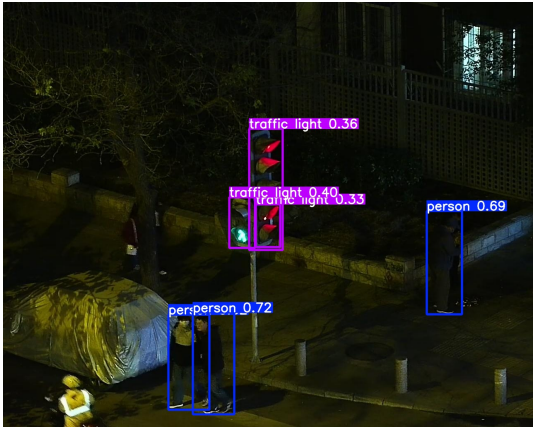
Model	2 dots	3 dots	4 dots	5 dots
Nano	59.90%	69.80%	77.79%	82.74%
Small	52.43%	61.47%	79.38%	72.99%
Medium	48.34%	59.57%	65.14%	71.78%

Table: Attack Comparison with Previous Works

Approach	**ASR**	**Efficiency Considerations**
AdvLB [3]	95.1%	No
AdvLS [4]	75.8%	No
Ours	82.74%	Yes

Attack Demonstration

Before Attack

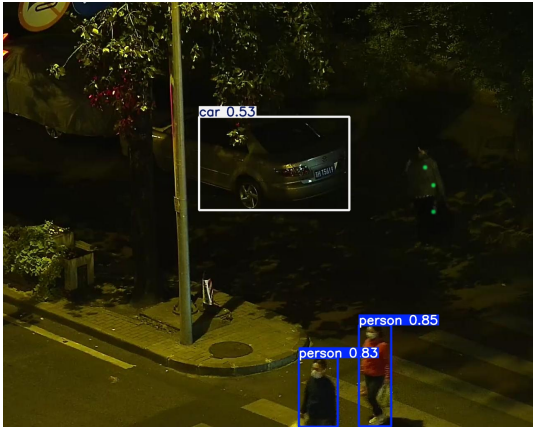


After Attack

Attack Demonstration

Before Attack

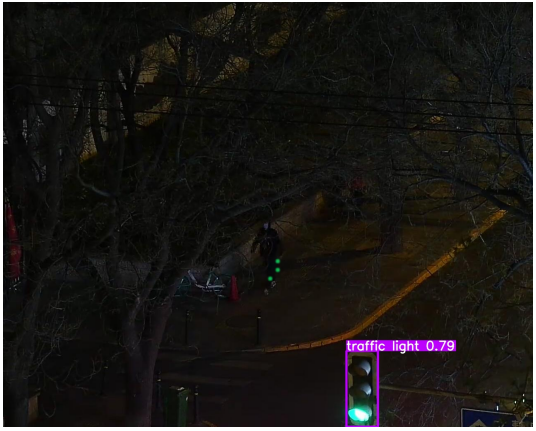
After Attack



Attack Demonstration

Before Attack

After Attack



Feasibility Analysis

- Real-time execution is crucial for the attack, requiring rapid optimization of dot placements.
- A feasibility study is necessary to evaluate the computational cost and ensure practical execution.

Current Hardware Specifications

- CPU: AMD Ryzen 7 (4000 series)
- GPU: NVIDIA GTX 1650

Attack Time Analysis

- We analyze attack times across different models and number of dots.
- Quantile heatmaps visualize completion times for 50%, 75%, 90%, and 95% of successful attacks.

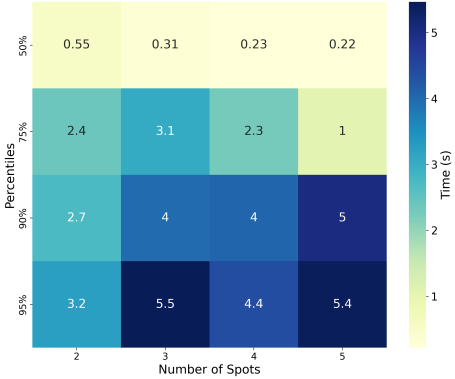


Figure 6.2: YOLO-Nano

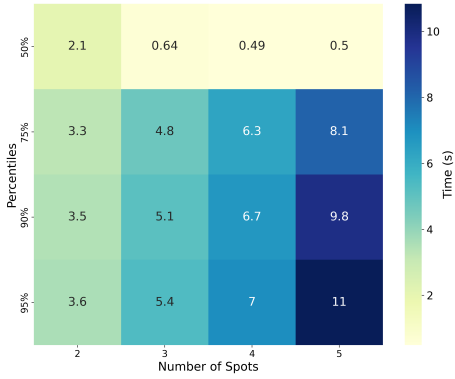


Figure 6.3: YOLO-Small

Attack Time Insights

- The attack is fast and practical for real-time use.
- Two key trends observed:
  1. **More dots = faster attacks:** For the same model size, increasing the number of laser dots reduces the time for 50% of attacks to succeed.
  2. **Larger models = slower attacks:** Bigger models increase the median attack time due to slower inference speeds.

Ablation Study

- We conduct experiments to analyze how various parameters affect the attack's effectiveness.
- This helps us understand the individual impact of different factors.
- Specifically, we study the effect of:
  1. **Wavelength**
  2. **Restarts**

Wavelength Impact Analysis

- We analyze how the laser's wavelength (λ) affects adversarial impact.
- Laser dots are placed randomly on pedestrians (non-optimized).
- Same dataset used, varying dot count from 2 to 5.
- Tested against Nano model, repeated 5 times per setup.
- Wavelengths: 380 to 680 nm (100 nm step), plus 532 nm (used in earlier tests).

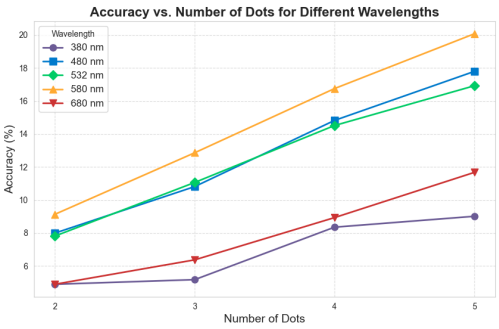


Figure: Effect of wavelength on detection success

Effect of Restarts on Attack Success

- We study the impact of varying the number of restarts (1 to 20) on attack performance.
- Experiments are conducted using YOLOv8-Nano and 3 laser dots on the same sampled dataset.
- As shown in Table 6.4, ASR improves with more restarts—from 63.83% at 1 restart to 75.76% at 20.
- More restarts help explore different initial conditions, increasing success.
- However, they also increase computation time—highlighting a trade-off between ASR and runtime.

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

Table 6.4: ASR vs. Number of Restarts

Conclusion

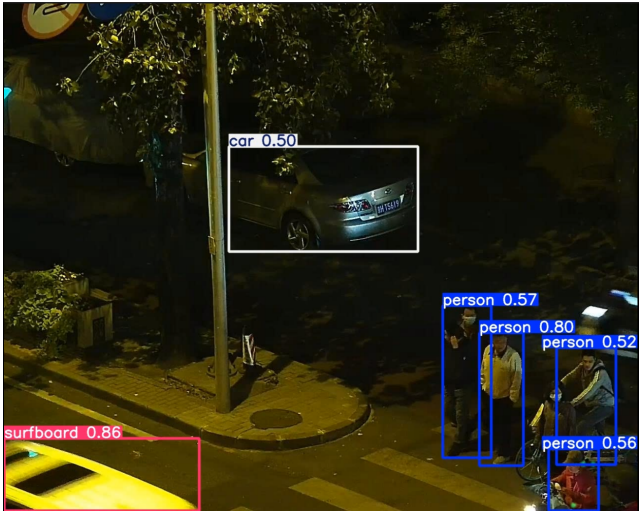
- We proposed a laser-based adversarial attack against pedestrian detection by projecting laser spots onto pedestrians to mislead object detectors.
- The attack was formulated as an optimization problem and implemented using a greedy search algorithm for real-time execution.
- We tested our method on YOLOv8-Nano, Small, and Medium models.
- Our approach achieved up to **82.7% ASR** with **5 laser spots**, completing **50% of attacks in under 0.22s**.
- The attack remained effective across different model sizes, with slightly reduced ASR on larger models due to increased robustness.
- Ablation studies on **wavelength** and **restart count** confirmed key performance trade-offs.

Future Work

- Develop real-time **defense mechanisms**, including:
  - Sensor-level filters
  - Model improvements
  - Anomaly detection strategies
- Improve attack **efficiency and speed**:
  - Faster search algorithms
  - Parallel computation
  - Use of high-performance GPUs
- Explore **learning-based approaches**, such as deep reinforcement learning or heuristic strategies for laser spot placement.
- Conduct tests in real-world physical settings and evaluate on diverse detection models to assess **robustness and generalizability**.

Bonus Video Demonstration

Output Video Demonstration



References I

[1] Tom Brown et al. “Adversarial Patch”. In: (Dec. 2017). DOI: 10.48550/arXiv.1712.09665.

[2] Yuanwan Chen et al. “Reflective Adversarial Attacks against Pedestrian Detection Systems for Vehicles at Night”. In: *Symmetry* 16.10 (2024). ISSN: 2073-8994. DOI: 10.3390/sym16101262.

[3] Ranjie Duan 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. 16062–16071.

[4] Chengyin Hu et al. “Adversarial laser spot: Robust and covert physical-world attack to dnns”. In: *Asian conference on machine learning*. PMLR. 2023, pp. 483–498.

References II

[5] Xinyu Jia et al. “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.

[6] Justin. *Autonomous Vehicle*. 2023. URL: https://media.wired.com/photos/6430be5f3715dbabc0357bd9/master/w_2240,c_limit/Autonomous-Vehicles-Are-Clogging-San-Francisco-Business-1390904981.jpg.

[7] Giulio Lovisotto et al. “SLAP: Improving Physical Adversarial Examples with Short-Lived Adversarial Perturbations”. In: *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, Aug. 2021, pp. 1865–1882. ISBN: 978-1-939133-24-3. URL: https://www.usenix.org/conference/usenixsecurity21/presentation/lovisotto.

[8] RenPang. *Fast gradient sign method*. 2020.

References III

[9] Tensorflow. *Fast gradient sign method*. 2024. URL: https://www.tensorflow.org/static/tutorials/generative/images/adversarial_example.png.

[10] Tsuruoka. *WIP: Adversarial Retroreflective Patches: A Novel Stealthy Attack on Traffic Sign Recognition at Night*. 2020.

[11] Kaidi Xu 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*. Glasgow, United Kingdom: Springer-Verlag, 2020, pp. 665–681. ISBN: 978-3-030-58557-0. DOI: 10.1007/978-3-030-58558-7_39.

Thank You

(Questions are welcome!)