CS5100 - FOUNDATIONS OF ARTIFICIAL INTELLIGENCE

# ADVERSARIAL ROBUSTNESS IN VISION-BASED ROBOT NAVIGATION

August 2025

Hemanth Raj Tekumalla[1]

Kodanda Rama Naidu Dumpala[2]

Master of Science in Robotics
Northeastern University

{tekumalla.h, dumpala.ko}@northeastern.edu

## Project Description

**Objectives:** This project investigates the vulnerability of modern object detection systems used in autonomous navigation to adversarial perturbations. We aim to quantify how simple perturbations can compromise safety-critical object detection and evaluate practical defenses suitable for embedded robotic systems.
**Scope:** We focus on YOLOv8 object detector using the nuScenes autonomous driving dataset, implementing black-box adversarial attacks and evaluating four preprocessing-based defense mechanisms under real-time constraints.

## Tools and Libraries Used

The project implementation leverages the following tools and frameworks:

- **Deep Learning:** PyTorch, Ultralytics YOLOv8
- **Computer Vision:** OpenCV, PIL/Pillow
- **Data Processing:** NumPy, Pandas
- **Visualization:** Matplotlib, Seaborn
- **Development:** Python 3, Jupyter Notebooks, Git

## Setup Instructions

To reproduce this project:

```
# Clone the repository
https://github.com/thraj0103/FAI_Final_Project.git
cd FAI_Final_Project

# Create virtual environment
python -m venv venv
source venv/bin/activate  # On Windows: venv\Scripts\activate

# Download nuScenes mini dataset
# Register at https://www.nuscenes.org/nuscenes
# Download and extract to data/v1.0-mini/

```

```
# Run experiments
jupyter notebook 01_dataset_preparation.ipynb
jupyter notebook 02_models_and_attacks.ipynb
jupyter notebook 03_defenses_and_evaluation.ipynb
```

# INTRODUCTION

The deployment of autonomous robots in human-populated environments represents a critical challenge in modern AI systems. These robots, ranging from delivery vehicles navigating sidewalks to self-driving cars on city streets, fundamentally depend on computer vision for safe navigation. At the core of this perception pipeline lies object detection, the ability to identify and localize pedestrians, vehicles, and obstacles that directly impact navigation decisions.

## Problem Statement

Modern object detection systems like YOLOv8 [1] achieve impressive accuracy on standard benchmarks, reporting mean Average Precision (mAP) scores exceeding 50% on challenging datasets. However, these metrics mask a critical vulnerability: susceptibility to adversarial perturbations, small, often imperceptible modifications to input images that cause catastrophic detection failures.

The problem becomes particularly acute when considering the constraints of deployed robotic systems. Unlike research environments with abundant computational resources, embedded systems must process sensor data in real-time (typically under 100ms), operate within strict power budgets, and often cannot be updated post-deployment due to safety certification requirements. Furthermore, real-world threats include both natural perturbations (sensor noise, compression artifacts) and potential malicious attacks, requiring defenses that work without knowledge of specific attack methods.

## Motivation and Goals

Recent demonstrations of physical adversarial attacks from patches causing traffic sign misclassification to road marking modifications inducing lane departure highlight the urgency of addressing these vulnerabilities. As robots increasingly share spaces with humans, ensuring robust perception becomes not just a technical challenge but an ethical one too.

This project pursues two primary goals:

**Goal 1: Quantify Real-World Vulnerability.** We systematically assess how adversarial perturbations affect safety-critical object detection in navigation contexts, focusing specifically on pedestrians, vehicles, and cyclists rather than generic accuracy metrics.

**Goal 2: Develop Practical Defenses.** We identify and evaluate defense mechanisms that satisfy embedded system constraints requiring no model retraining, minimal computational overhead, and implementation using existing hardware capabilities.

# BACKGROUND

## Related Work

Adversarial examples were first discovered by Szegedy et al. (2013) [2], who showed that imperceptible perturbations could fool neural networks. Goodfellow et al. (2014) [3] provided theoretical grounding through the Fast Gradient Sign Method (FGSM), establishing the $L_\infty$ threat model where perturbations are bounded by maximum per-pixel changes.

While extensive research exists on adversarial attacks for image classification, object detection presents unique challenges. Xie et al. (2017) [4] proposed Dense Adversary Generation targeting multiple objects simultaneously, while Liu et al. (2018) [5] introduced DPatch, creating universal patches that suppress detection. Particularly relevant to navigation, Eykholt et al. (2018) [6] demonstrated physical attacks on traffic sign detection.

Defense strategies broadly fall into three categories: adversarial training (computationally expensive), certified defenses (requiring specialized architectures), and preprocessing defenses. For embedded systems, preprocessing approaches like JPEG compression (Dziugaite et al., 2016) [7] and Gaussian filtering (Li et al., 2017) [8] offer practical solutions requiring no model modification.

## AI/ML Concepts Used

This project leverages several key AI concepts:

**Object Detection:** Unlike classification that assigns single labels, detection must simultaneously localize and classify multiple objects. YOLOv8 [1] uses a single-stage architecture dividing images into grids and directly predicting bounding boxes and class probabilities.

**Adversarial Perturbations:** These exploit the high-dimensional nature of neural network input spaces, where small changes can cross decision boundaries. In detection, perturbations can target confidence thresholds, non-maximum suppression, or multi-scale features.

# METHODOLOGY

## Tools and Frameworks

We implemented our experiments using PyTorch [9] for tensor operations, Ultralytics YOLOv8 [1] for object detection, and OpenCV [10] for image preprocessing. All experiments ran on an M1 MacBook Pro.

## Data Sources and Preprocessing

We utilized the nuScenes mini dataset containing 400 front-camera images from urban driving scenarios. This dataset provides diverse lighting conditions, weather variations, and complex urban environments representative of real navigation challenges. Figure 1 shows example images from our dataset, demonstrating the variety of scenarios encountered in autonomous navigation.

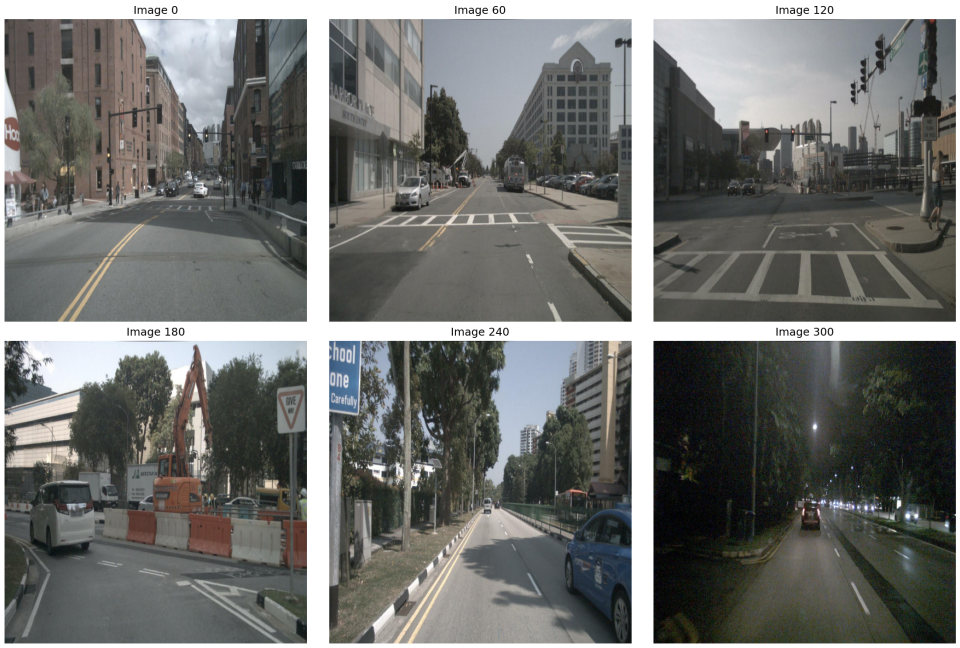


**Figure 1:** Sample images from nuScenes dataset showing diverse urban navigation scenarios including day/night conditions, construction zones, and varying traffic densities

Images underwent standardized preprocessing: resizing to 640×640 pixels (YOLOv8's expected input), converting BGR to RGB color space, and normalizing pixel values to [0,1]. This maintains visual information while ensuring consistency across experiments.

### Algorithms and Models

**Targeted Random Perturbation Attack:** Our black-box attack iteratively searches for perturbations that suppress critical object detection:

```
def targeted_attack(image, epsilon, num_tries=10):
    original = image.copy()
    orig_count = count_critical_objects(model(original))

    best_perturbed = original
    min_detections = orig_count

    for _ in range(num_tries):
        # Random perturbation within L-infinity bound
        noise = np.random.uniform(-epsilon, epsilon, image.shape)
        perturbed = np.clip(original + noise, 0, 1)

        # Keep if fewer objects detected
        new_count = count_critical_objects(model(perturbed))
        if new_count < min_detections:
            min_detections = new_count
            best_perturbed = perturbed

    return best_perturbed
```

**Defense Mechanisms:** We evaluated four preprocessing defenses:

- **Gaussian Blur:** 3×3 kernel smoothing high-frequency perturbations
- **JPEG Compression:** Quality factor 75 removing imperceptible details
- **Bit Reduction:** Quantizing 8-bit to 5-bit representation
- **Random Resize:** Scaling to 90% then padding to break spatial correlations

## RESULTS

### Model Evaluation Metrics

Our evaluation focused on three key metrics tailored to navigation safety:

- **Critical Object Suppression Rate:** Percentage of safety-critical objects (persons, vehicles, bicycles) that become undetected under adversarial perturbations.

- **Defense Recovery Rate:** Percentage of suppressed detections restored by each defense mechanism.
- **Computational Overhead:** Processing time in milliseconds, with <10ms considered suitable for embedded deployment.

## Attack Effectiveness

Our targeted random perturbation attack achieved substantial suppression across tested epsilon values, as shown in Figure 2. The relationship between perturbation magnitude and detection suppression demonstrates concerning vulnerability even at low epsilon values.

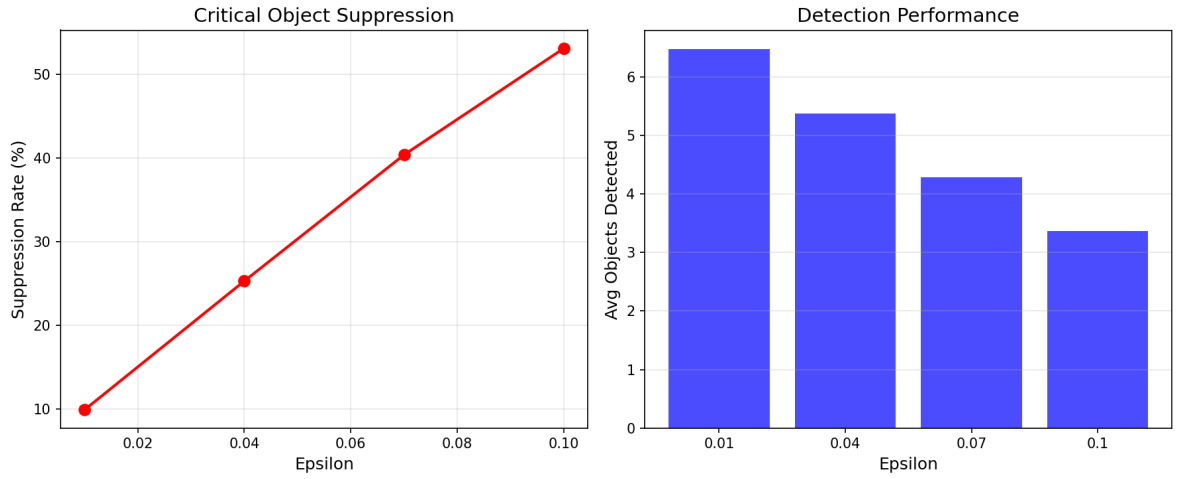


**Figure 2:** Attack effectiveness: (a) Critical object suppression rate increases with epsilon, reaching over 50% at $\epsilon = 0.1$. (b) Average number of critical objects detected decreases from 6.5 to 3.4 as epsilon increases

Key findings from our attack evaluation:

- $\epsilon = 0.01$: 9.9% suppression (imperceptible to humans)
- $\epsilon = 0.04$: 25.3% suppression (barely perceptible)
- $\epsilon = 0.07$: 40.4% suppression (noticeable if examined closely)
- $\epsilon = 0.10$: 53.1% suppression (clearly visible noise)

Figure 3 provides a visual demonstration of how increasing perturbation strength progressively degrades object detection performance on a real navigation scene.

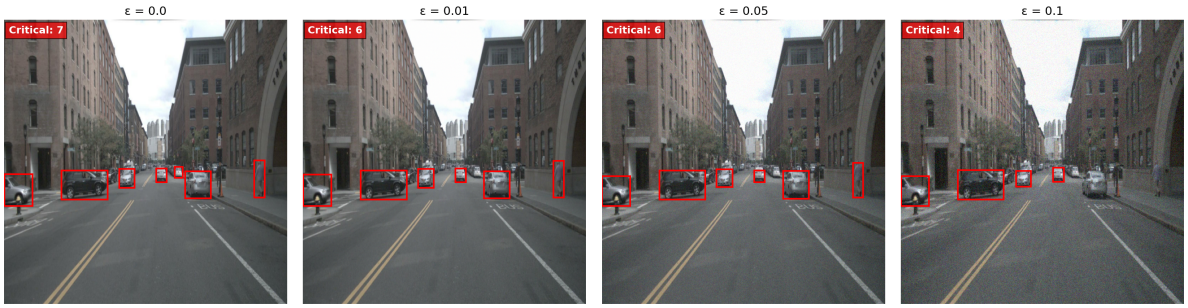


**Figure 3:** Visual demonstration of attack effects at different epsilon values. Red boxes indicate detected critical objects. As epsilon increases from 0.0 to 0.1, the number of detected objects decreases from 7 to 4, with several vehicles becoming undetected

## Defense Performance

Against attacks with $\epsilon = 0.03$, our evaluated defenses showed varying effectiveness in recovering suppressed detections. Figure 4 illustrates how different preprocessing defenses restore detection capability on adversarially perturbed images.


**Figure 4:** Visual comparison of defense mechanisms. Original clean image detects 7 objects, adversarial image (none) detects 6, while blur and resize defenses recover to 7 detections. JPEG and bit reduction show partial recovery

Figure 5 provides a comprehensive visualization of defense performance, clearly showing Gaussian blur's superiority in both recovery rate and practical deployability.

Defense Effectiveness and Computational Overhead

| Defense | Recovery Rate | Overhead (ms) | Embedded Suitable |
|---|---|---|---|
| None (Baseline) | 0.0% | 0.42 | Yes |
| JPEG Compression | 45.4% | 19.39 | No |
| Gaussian Blur | **62.9%** | 8.46 | Yes |
| Bit Reduction | 27.4% | 5.27 | Yes |
| Random Resize | 49.6% | 7.56 | Yes |

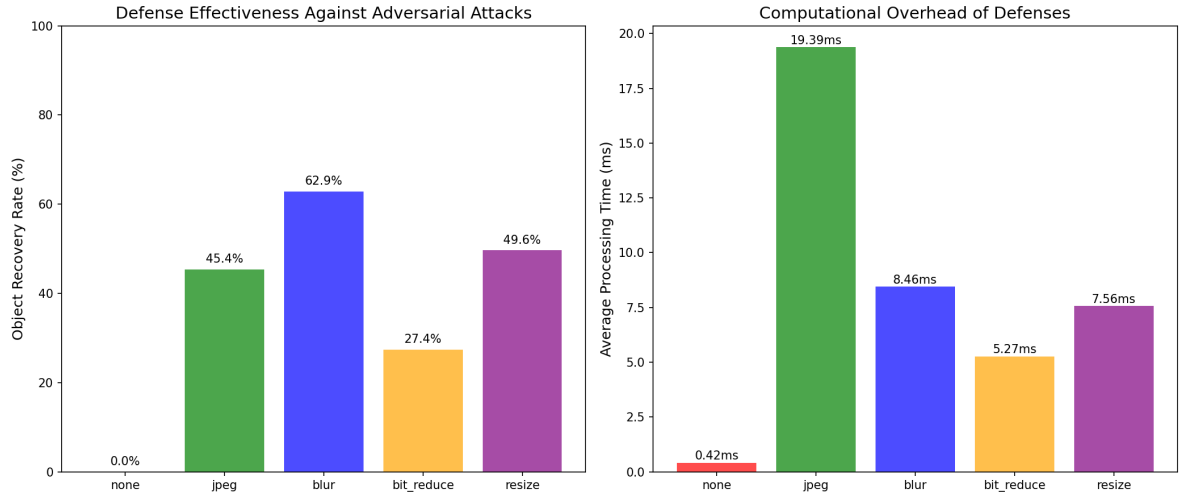


**Figure 5:** Defense effectiveness comparison: (a) Object recovery rates showing Gaussian blur achieving 62.9% recovery. (b) Computational overhead where only JPEG compression exceeds the 10ms threshold for embedded systems

Gaussian blur emerged as a very good defense, achieving the highest recovery rate while maintaining acceptable overhead for real-time operation. The 8.46ms processing time fits comfortably within typical perception budgets of 30-50ms in robotic systems. The practical impact of our findings becomes clear when examining specific detection scenarios. In urban navigation contexts, missing critical objects like pedestrians or vehicles can have catastrophic consequences. Our experiments demonstrate that simple preprocessing can significantly improve robustness without requiring model modifications or retraining.

# DISCUSSION

## Interpretation of Results

The demonstrated vulnerability to simple random perturbations reveals a critical safety gap in deployed vision systems. A 25% suppression rate at barely perceptible perturbation levels means one in four critical objects might go undetected—potentially catastrophic in navigation contexts.

Gaussian blur's performance stems from its low-pass filtering properties, effectively removing high-frequency adversarial noise while preserving object features. The 62.9% recovery rate, combined with 8.46ms overhead fitting within typical perception budgets (30-50ms), makes it practically deployable without hardware modifications.

## Limitations

Our evaluation focused on digital perturbations without testing physical-world attacks. We examined only YOLOv8, though other architectures might exhibit different vulnerabilities. The static nature of defenses cannot adapt to evolving attacks, and our relatively small dataset may not capture all real-world scenarios.

## Potential Improvements

Future work should evaluate transferability across detector architectures, implement adaptive defenses adjusting to threat levels, and integrate findings with multi-sensor fusion for defense-in-depth. Physical-world validation using printed patches or projected patterns would strengthen practical applicability.

# CONCLUSION

This project successfully demonstrated that modern object detectors exhibit vulnerabilities to adversarial perturbations, with simple random noise suppressing up to 53% of critical detections at visible perturbation levels. More concerning, even barely perceptible perturbations achieved 25% suppression rates, revealing a fundamental fragility in vision-based navigation systems. Through systematic evaluation, we identified Gaussian blur as one of the optimal defenses for embedded deployment, recovering 62.9% of suppressed detections

while maintaining real-time performance with only 8.46ms overhead. The investigation reinforces that standard accuracy metrics provide false confidence for safety-critical applications, where worst-case performance under adversarial conditions matters more than average-case benchmarks. As autonomous systems increasingly share spaces with humans, our work contributes both a methodology for adversarial evaluation and immediately deployable defenses. The key lesson extends beyond technical contributions: safe autonomous navigation demands explicit consideration of adversarial robustness in development and certification processes, treating it not as an academic curiosity but as a fundamental safety requirement comparable to crash testing in automotive design.

# Bibliography


[1] 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*, pp. 779–788, 2016.

[2] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

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

[4] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille, "Adversarial examples for semantic segmentation and object detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 1369–1378, 2017.

[5] X. Liu, H. Yang, Z. Liu, L. Song, H. Li, and Y. Chen, "Dpatch: An adversarial patch attack on object detectors," *arXiv preprint arXiv:1806.02299*, 2018.

[6] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning visual classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1625–1634, 2018.

[7] G. K. Dziugaite, Z. Ghahramani, and D. M. Roy, "A study of the effect of jpg compression on adversarial images," *arXiv preprint arXiv:1608.00853*, 2016.

[8] X. Li and F. Li, "Adversarial examples detection in deep networks with convolutional filter statistics," in *Proceedings of the IEEE international conference on computer vision*, pp. 5764–5772, 2017.

[9] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[10] Itseez, "Open source computer vision library." `https://github.com/itseez/opencv`, 2015.