ADA Project

Robustness Assessment of YOLOv9e Model Against Black-Box Adversarial Attacks

Madhav Sood IMT2021009

Yukta Rajapur IMT2021066

Brij Desai IMT2021067

Shlok Agrawal IMT2021103

October 27, 2025

Table of Contents

1 Summary **1**

2 System and Dataset Background **2**

2.1 The Model: YOLOv9e 1

2.2 The Dataset: DOTA 2

3 Baseline Performance Assessment **2**

4 Adversarial Attack Methodology **3**

4.1 One-Pixel Random Search (1-px RS) 3

4.2 One-Pixel Differential Evolution (1-px DE) 3

5 Results & Analysis **5**

5.1 Analysis of One-Pixel Attacks (RS vs. DE) 5

1 Summary

This report details an investigation into the adversarial robustness of a state-of-the-art YOLOv9e object detection model. The model, trained on the DOTA aerial imagery dataset, was subjected to a series of escalating black-box adversarial attacks.

A performance baseline was first established on the clean validation set, yielding a mAP@0.50 of 0.7174.

The model was then subjected to two distinct one-pixel attacks: one using a simple Random Search (RS) and another using an intelligent Differential Evolution (DE) algorithm. Both attacks failed to significantly degrade the model's performance. The RS attack resulted in a mAP@0.50 of 0.7144, and the DE attack resulted in a nearly identical mAP@0.50 of 0.7143, both representing a negligible performance drop of less than 0.5

The investigation was then escalated to a three-pixel Differential Evolution (3-px DE) attack that was executed on a random 5% sample of the validation set (18 images) for compute reasons. On the sampled set the 3-pixel DE attack reduced mAP@0.50 from 0.6172 to 0.6100 (drop = 0.0071), indicating only a very small degradation at this sample size.

The primary conclusion from the initial experiments is that the YOLOv9e model exhibits exceptionally high robustness to minimal, single-pixel perturbations.

The GitHub repository with our code: https://github.com/brij-desaii/Adverserial_Attacks_On_DOTA.

2 System and Dataset Background

2.1 The Model: YOLOv9e

The model under investigation is YOLOv9e. YOLO (You Only Look Once) models are renowned for their exceptional balance of real-time processing speed and high accuracy. The YOLOv9 architecture introduced novel concepts such as Programmable Gradient Information (PGI) and the Generalized Efficient Layer Aggregation Network (GELAN), which allow it to learn more comprehensive information and achieve a superior accuracy-efficiency trade-off. The 'e' (extended) variant is one of the larger, more powerful models in the series, making it a robust and challenging target for this study.

2.2 The Dataset: DOTA

The model was trained on the DOTA (Dataset for Object Detection in Aerial Images). This is a large-scale and highly challenging dataset for object detection, characterized by:

- Massive Image Size: Images are often as large as 4000x4000 pixels.
- High Object Density: Images frequently contain a high density of objects.
- Scale Variation: Objects range from very large (e.g., bridges) to very small (e.g., small vehicles).

Our model was trained on 16 classes, including plane, ship, storage tank, large-vehicle, and small-vehicle. The validation set used for all experiments consists of 374 images.

3 Baseline Performance Assessment

Before conducting any attacks, we established the model's baseline performance on the clean, un-attacked validation set. This provides the "ground truth" metric against which all attack effectiveness is measured.

The primary metrics are:

- mAP@0.50: Mean Average Precision at an IoU (Intersection over Union) threshold of 50%. This is our main indicator of performance.
- Precision: Of all the detections the model made, what percentage was correct?
- Recall: Of all the true objects in the image, what percentage did the model find?

Baseline Results (on Clean Data):

- mAP@0.50: 0.7174
- mAP@0.50:0.95: 0.5629
- Precision: 0.8770
- Recall: 0.6462

4 Adversarial Attack Methodology

All experiments share a common black-box methodology. The core of this is a fitness function (or "badness" score) that we aim to minimize. This function is defined as the sum of all detection confidence scores for a given image. A lower score indicates the model is either detecting fewer objects or is less confident about its detections, both of which are desirable outcomes for the attacker.

4.1 One-Pixel Random Search (1-px RS)

Objective: To find the single most disruptive pixel using a simple, brute-force random search.

Algorithm:

1. The original image was fed to the model to get its base_score.
2. A loop was initiated for 30 attempts.
3. In each attempt, a random pixel coordinate (x, y) was selected.
4. At this location, four distinct color modifications were tested by creating four copies of the image:
   - invert: The pixel's (B, G, R) color was inverted (255 - color).
   - zero: The pixel was set to black (0, 0, 0).
   - max: The pixel was set to white (255, 255, 255).
   - random: The pixel was set to a new, random (B, G, R) value.
5. Each of these four modified images (totaling 120 model queries per image) was passed to the model, and its score was recorded.
6. The single pixel modification (a specific coordinate and color mode) that resulted in the lowest score was saved as the final attacked image.

4.2 One-Pixel Differential Evolution (1-px DE)

Objective: To replace the inefficient random search with an intelligent evolutionary algorithm to find the optimal one-pixel attack. This attack is defined by a 5-dimensional vector: (x_coord, y_coord, red_val, green_val, blue_val).

Reference: This method is based on the "Differential Evolution" algorithm by Storn and Price (1997).

Algorithm:

1. A "population" of 10 random candidate vectors (10 different pixel attacks) was initialized.

2. The "fitness" (our "badness" score) of each candidate was calculated.

3. The algorithm then "evolved" this population for 25 generations. In each generation, it created new "offspring" candidates by combining and mutating the parameters of successful parents.

4. If an offspring was "fitter" (i.e., produced a lower score) than its parent, it replaced the parent in the next generation.

5. This process intelligently converges on an optimal or near-optimal solution. The best candidate found across all generations was saved as the final attack.

4.3 Three-Pixel Differential Evolution (3-px DE)

Objective: To escalate the attack potency by jointly optimising three distinct pixel modifications (three coordinates and three RGB values per pixel) using a black-box Differential Evolution (DE) search to minimise the model's aggregate detection confidence score (the "badness" score).

Algorithm / Implementation details:

1. **Attack encoding:** each candidate solution is a 15-dimensional vector

$$(x_1, y_1, r_1, g_1, b_1,\ x_2, y_2, r_2, g_2, b_2,\ x_3, y_3, r_3, g_3, b_3),$$

representing three pixel locations and their BGR values.

2. **Fitness function:** for a given candidate we modify the original image at the three specified pixel locations and evaluate the model; the fitness is the sum of all detection

confidence scores returned by the model (the optimisation goal is to *minimise* this value).

3. **DE configuration (reported experiments):** `maxiter=30`, `popsize=20`, `seed=42`, `workers=1`. Pixel coordinates are constrained to image bounds and RGB values to $[0, 255]$.

4. **Sampling and practical constraints:** because full-dataset DE is computationally heavy, we ran the 3-px attack on a random 5% sample of the validation set (18 images) — the same sampling procedure used to produce baseline predictions for comparability. For each sampled image the DE run returns the best three-pixel perturbation found and an attacked image is saved for downstream evaluation.

5 Results & Analysis

Table 1: 1-px Performance Results

Metric	**Baseline (Clean)**	**1-px Random Search**	**1-px DE**
mAP@0.50	**0.7174**	**0.7144**	**0.7143**
Precision	0.8770	0.8395	0.8380
Recall	0.6462	0.6562	0.6566
mAP@0.50:0.95	0.5629	0.5549	0.5550

Table 2: Sampled 3-px Performance Results

Metric	Sampled baseline (5%)	3-px DE (sampled 5%)
mAP@0.50	0.6172	0.6100
mAP@0.50:0.95	0.4320	0.4313
Precision	0.8332	0.8265
Recall	0.5176	0.5122

5.1 Analysis of One-Pixel Attacks (RS vs. DE)

The results from the first two experiments are clear: the one-pixel attack, in both forms, failed.

The mAP drop was a negligible 0.4%, which is statistically insignificant. This demonstrates that the model is exceptionally robust to such minimal perturbations.

The most interesting finding is the **Precision-Recall Anomaly**:

- Precision Dropped: Both attacks caused a noticeable drop in Precision (RS: -4.3%, DE: -4.4%). This indicates the attacks were successful in causing the model to generate more false positives (hallucinating objects that were not there).
- Recall Increased: This drop was, however, completely offset by a surprising increase in Recall (RS: +1.5%, DE: +1.6%). This suggests that the random pixel perturbations, by pure chance, also made some hard-to-detect, real objects easier for the model to see.

Furthermore, the DE search performed identically to the random search. This implies that the 1-pixel problem space is too simple; there is no complex, hidden vulnerability for DE to discover, and any random pixel has a similarly small and unpredictable effect.

5.2 Analysis of Three-Pixel Attack

The three-pixel DE escalation produced only a marginal degradation on the sampled validation subset. On the 5% sample (18 images) the model's mAP@0.50 decreased from 0.6172 (sampled baseline) to 0.6100 after the 3-px DE attack (drop = 0.0071). Other metrics show similarly small changes: mAP@0.50:0.95 fell from 0.4320 to 0.4313, Precision from 0.8332 to 0.8265, and Recall from 0.5176 to 0.5122. The mean per-image proxy score drop (base_score – attacked_score) across successful cases was **0.1613**, indicating that the DE search did find per-image reductions in model confidence, but these per-image reductions did not translate into a large aggregate degradation across the sampled set.

Interpretation: these results suggest that — for the sampled subset and with the DE hyperparameters used (`maxiter=30`, `popsize=20`) — the YOLO model remains resilient to sparse, three-pixel perturbations. The attack is capable of lowering confidence on an image-by-image basis, but the effect is small when averaged and when measured with standard detection metrics (mAP / precision / recall).

Practical note on compute: the per-image DE runs were non-trivial; the full sampled run (18 images, DE `maxiter=30`) required on the order of a couple minutes per image on the GPU used for experiments (Tesla T4), which motivated the sampling strategy.

Caveat and recommendation

The 3-px DE experiment was performed on a 5% random sample of the validation set (18 images) to bound compute and wall-clock time. Because the sample size is small, these results should be treated as indicative rather than definitive. To fully characterise the model's sensitivity to sparse multi-pixel attacks we recommend:

1. Repeat the 3-px DE evaluation on a larger random sample (e.g., 20–30% of validation) or the full validation set.

2. Consider increasing the DE budget (more generations / larger population) to allow stronger perturbations to be found.