Generating Minimalist Adversarial Perturbations to Test Object-Detection Models

Cristopher McIntyre-Garcia, Adrien Heymans, Beril Borali, Won-Sook Lee and Shiva Nejati

University of Ottawa

Background

Brendel et al. [3] propose gradient-based algorithms to assess model robustness by generating adversarial examples and measuring the time and noise needed for successful attacks. GenAttack [1] uses a genetic algorithm for generating black-box adversarial attacks on classification models. Bartlett et. al. [2] extends GenAttack by introducing a multi-objective algorithm for adversarial testing of classification models. EvoAttack [5] builds upon GenAttack, adapting it for adversarial attacks on object-detection models. This adaptation uses the aggregation of confidence scores from detected objects in an image as a fitness function.

Idea

Our work is similar to EvoAttack in performing adversarial attacks on object-detection models. We enhance this approach by introducing a multi-metric fitness measure that evaluates detection evasion and perturbation minimization without sacrificing runtime efficiency. The noise reduction measure is complemented by our adaptive fitness function. This function dynamically balances confidence scores that capture the effectiveness of attacks with noise reduction.

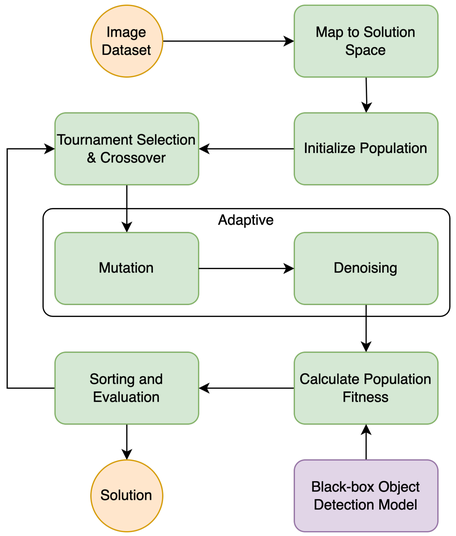

Fig. 1: TM-EVO Block Diagram

We have conducted an empirical evaluation involving both Convolutional Neural Networks (CNN)-based (Faster R-CNN) and Transformer-based (DETR) object detection models under test (MUT).

Multi-metrics function

Our multi-metric function combines a set of metrics to ensure an optimal amount of noise is applied to adversarial examples. M_1 computes the average confidence scores of all objects detected in I by the MUT:

$$M_1(I) = \frac{\sum_{i=0}^{n} conf_i}{n} \tag{1}$$

M_2 computes the number of pixels modified in I compared to I_{ori} divided by their total number of pixels:

$$M_2(I) = \frac{pixel_{mut}}{pixel_{total}} \tag{2}$$

M_3 calculates the Euclidean distance between the original image I_{ori} and the generated image I. This distance is normalized by the maximum Euclidean distance to a uniform image I_{uni}, either completely black or white, depending on which yields the larger $||\cdot||2$ with I_{ori}:

$$M_3(I) = \frac{\|I_{ori} - I\|_2}{\|I_{uni}\|_2} \tag{3}$$

All the above three metrics are normalized so as not to overpower one another. The fitness function is then defined as the weighted sum of the above three metrics:

$$fitness(I) = w_1 \cdot M_1(I) + w_2 \cdot M_2(I) + w_3 \cdot M_3(I) \tag{4}$$

Training

We use DETR [4] and Faster R-CNN [9] as object detecters. Faster R-CNN is based on a convolutional neural network (CNN) that has been applied to real-time applications like autonomous driving [9]. DETR uses a CNN backbone for feature extraction, and further, incorporates a transformer encoder-decoder architecture to detect objects [4]. We use the COCO [8] and KITTI [6] datasets. The COCO dataset is widely used in computer vision, it contains diverse images annotated with object instances across multiple categories. The KITTI dataset includes images related to autonomous driving cars. KITTI benchmarks TM-EVO's effectiveness in autonomous driving object-detection models.

Results

Figure 2 presents the results comparing the performance of TM-EVO and EVO. Note that both TM-EVO and EVO successfully found an attack for all the input images in all runs. Hence, in Figure 2, we focus on L0 and L2 norm and execution time results to compare TM-EVO and EVO in efficiently generating stealthy adversarial attacks.

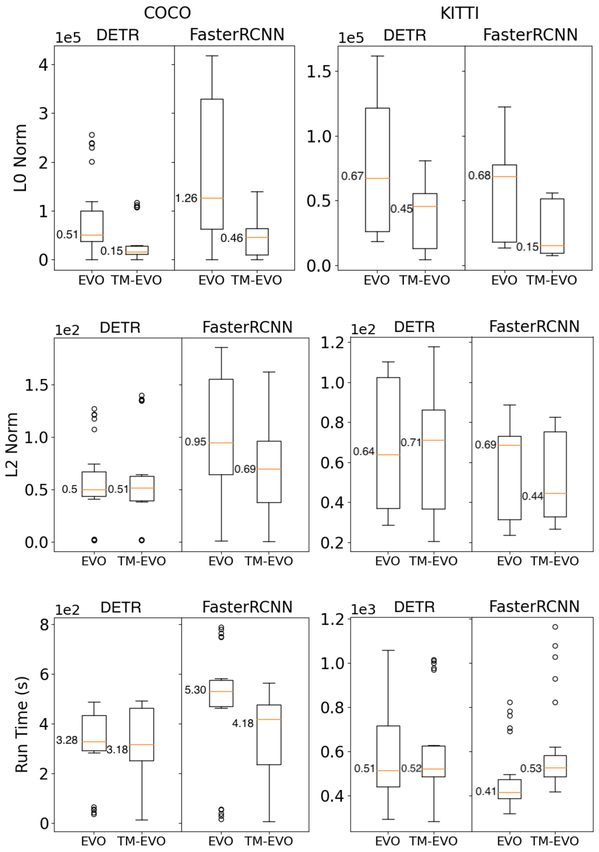

Fig. 2: Comparing TM-EVO and EVO algorithms: (a) L0 norm results, (b) L2 norm results, and (c) runtime results.

TM-EVO outperforms EVO in reducing noise in the generated attacks as measured by L0 and L2 norm metrics across both object-detection models and datasets examined in our study. On average, TM-EVO reduces noise in the generated attacks by 60.8% compared to EVO, as measured by the L0 norm metric. For Faster R-CNN, TM-EVO improves L2 norm results by an average of 30.9% over EVO for both datasets. This improvement is achieved without a notable rise in average execution time. Comparing DETR and Faster R-CNN yields interesting contrasts: In the case of DETR, the L0 and L2 norm results from EVO and TM-EVO are more similar compared to those for Faster R-CNN, where TM-EVO significantly outperforms EVO. In particular, with DETR, TM-EVO outperforms EVO in terms of L0 only. However, for Faster R-CNN, TM-EVO outperforms EVO in both noise-measurement metrics. This implies a limited potential for minimizing the degree of image perturbation for attacks generated for DETR, which uses a Transformer architecture, compared to those generated for Faster R-CNN's CNN-based framework.

Not So Loud!

Our method demonstrates better performance in the generation of adversarial attacks, which is consistently observed across both the KITTI and COCO datasets, as illustrated in Figure 3. This advancement not only highlights the robustness and adaptability of our approach but also its effectiveness in navigating through a wide array of object detection scenarios.

Fig. 3: EvoAttack vs. TM-EVO Qualitative Evaluation

Figure 3 contrasts the impact of two adversarial attack algorithms on the Faster R-CNN object detection model using COCO and KITTI datasets. The top images from COCO show that EVOAttack (left) introduces more visual noise compared to TM-EVO (right), which is less intrusive. Similarly, on the KITTI dataset, EVOAttack (bottom left) causes noticeable image distortion, while TM-EVO (bottom right) maintains clearer imagery with minimal distortion. The comparison highlights TM-EVO's ability to craft more subtle adversarial images that could potentially be more challenging for object detection models to identify.

Conclusion

Our work demonstrates the potential of multi-metric evolutionary search in achieving adversarial attacks with minimal noise. Moreover, we leverage the adaptability of evolutionary search algorithms to introduce measures that finely tune the generation of adversarial attacks and the required noise levels. Our adaptive, multi-metric evolutionary search approach has shown overall better results than the baseline approach.

In future work, we aim to explore alternative adaptation strategies and metrics to enhance and refine TM-EVO. In addition, we will conduct more empirical experiments to more effectively assess TM-EVO's efficiency and effectiveness. Our replication package is available online [7].

References

[1] M. Alzantot et al. "Genattack: Practical black-box attacks with gradient-free optimization". In: *GECCO'19*. 2019, pp. 1111–1119.

[2] A. Bartlett, C. Liem, and A. Panichella. "On the Strengths of Pure Evolutionary Algorithms in Generating Adversarial Examples". In: *SBFT'16 Workshop*. 2023.

[3] W. Brendel et al. "Accurate, reliable and fast robustness evaluation". In: *Advances in neural information processing systems* 32 (2019).

[4] N. Carion et al. "End-to-end object detection with transformers". In: *ECCV'20*. 2020, pp. 213–229.

[5] K. Chan and B. HC Cheng. "EvoAttack: An Evolutionary Search-Based Adversarial Attack for Object Detection Models". In: *SSBSE'22*. 2022, pp. 83–97.

[6] A. Geiger et al. "Vision meets robotics: The KITTI dataset". In: *International Journal of Robotics Research* 32.11 (2013), pp. 1231–1237.

[7] *Github repo for the paper.* `https://github.com/cmcin019/TM-EVO/tree/main`. [Online; accessed Jan 27, 2024]. 2023.

[8] Tsung-Yi Lin et al. *Microsoft COCO: Common Objects in Context*. 2015. arXiv: `1405.0312 [cs.CV]`.

[9] S. Ren et al. "Faster R-CNN: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems* 28 (2015).