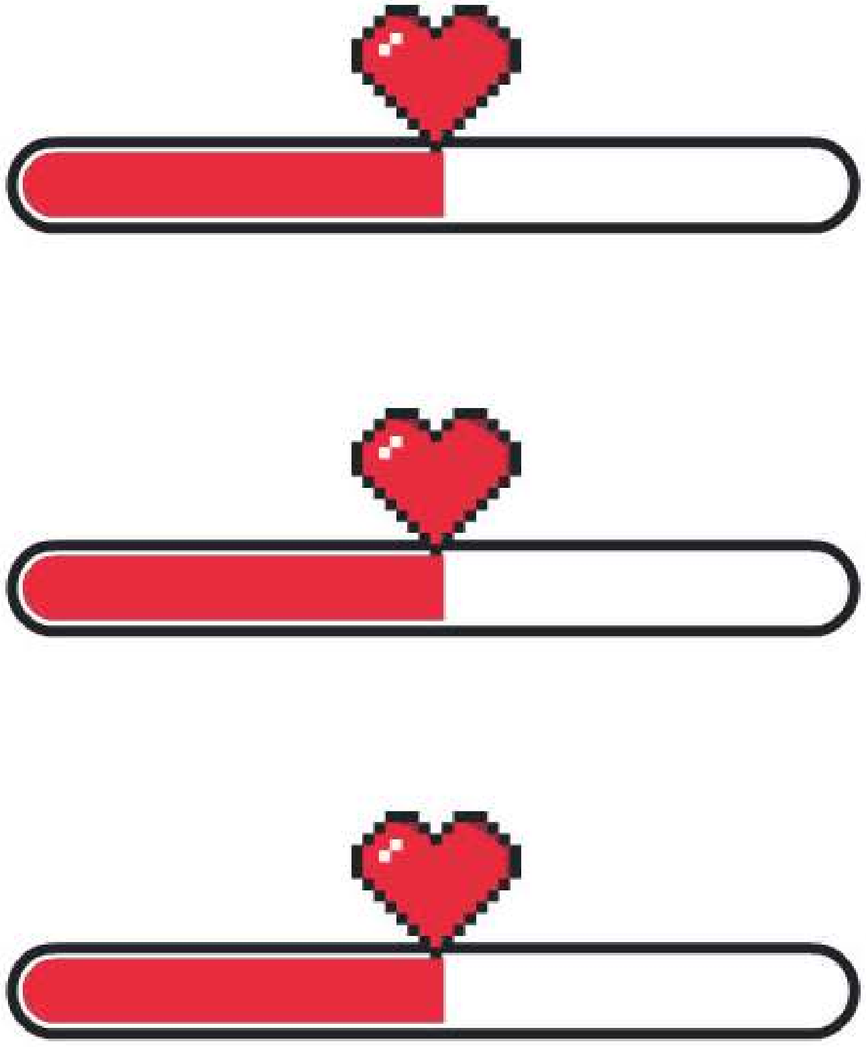



HACKER?

TẤN CÔNG ĐỐI KHÁNG

TRÊN MÔ HÌNH NHẬN DIỆN VẬT THỂ

SEAS

Mục lục

Our team

Nguyên Khôi
(mentor)

Minh Đức
(refrigerator)

Our team

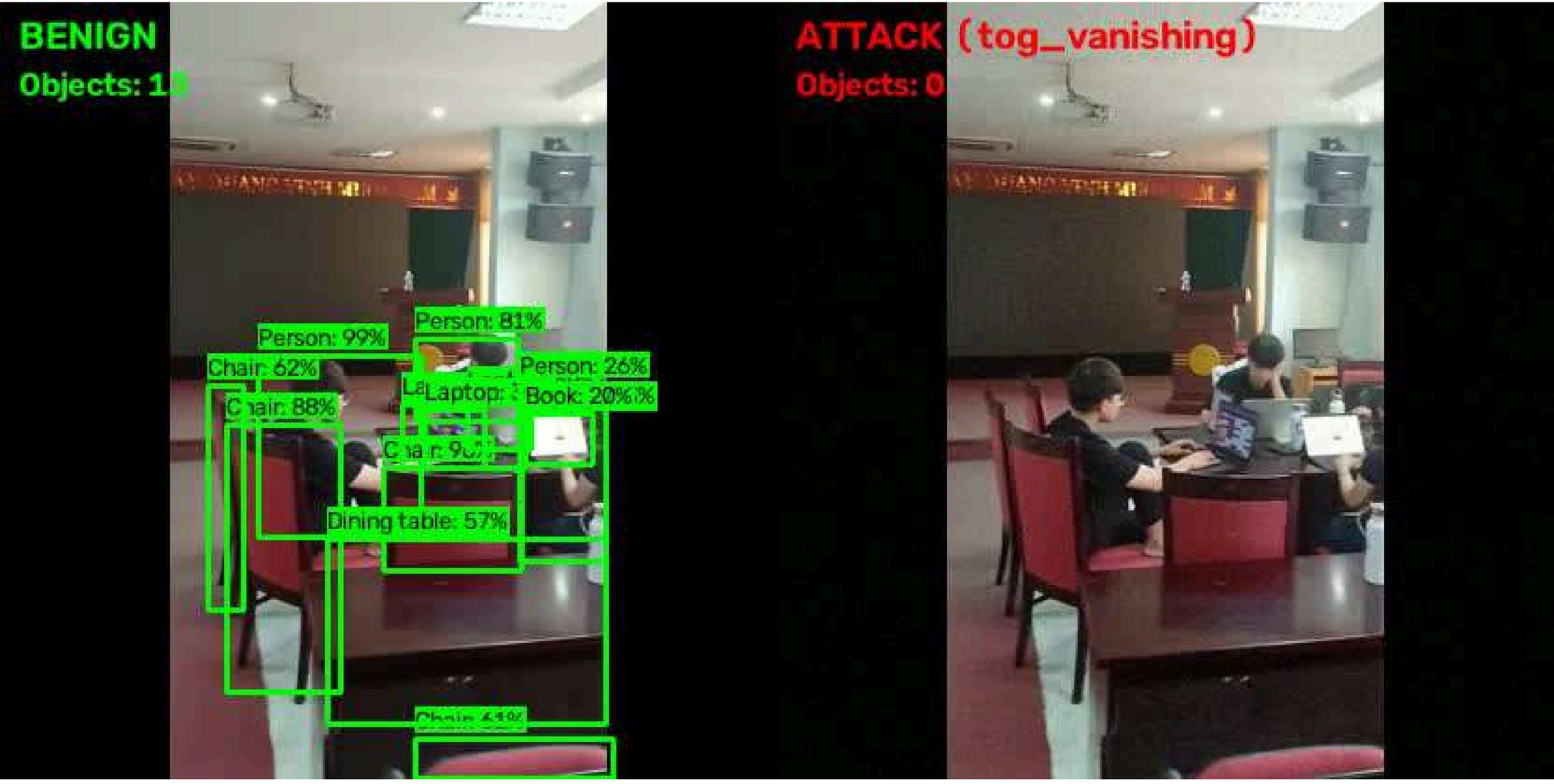
Phương Đông

Xuân Quân

Hoài Thương

Bình Minh

Quang Huy


BENIGN
Objects: 1
Person: 99%
Person: 81%
Chair: 62%
Person: 26%
Laptop:
Book: 20%
Dining table: 57%
ATTACK (tog_vanishing)
Objects: 0

Tầm quan trọng

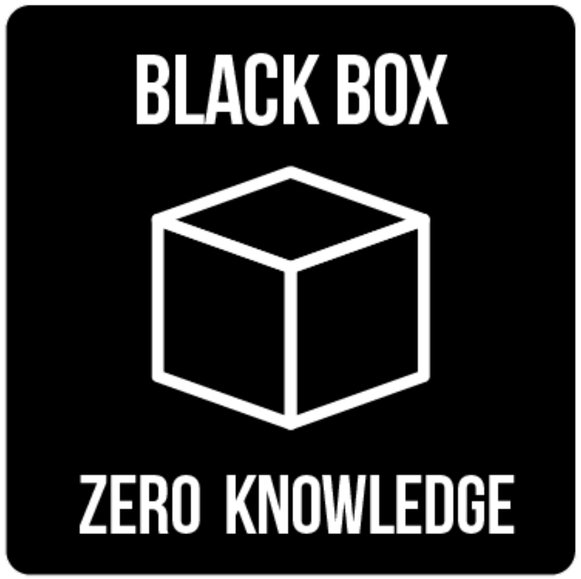
Đảm bảo An toàn cho các hệ thống thực tế

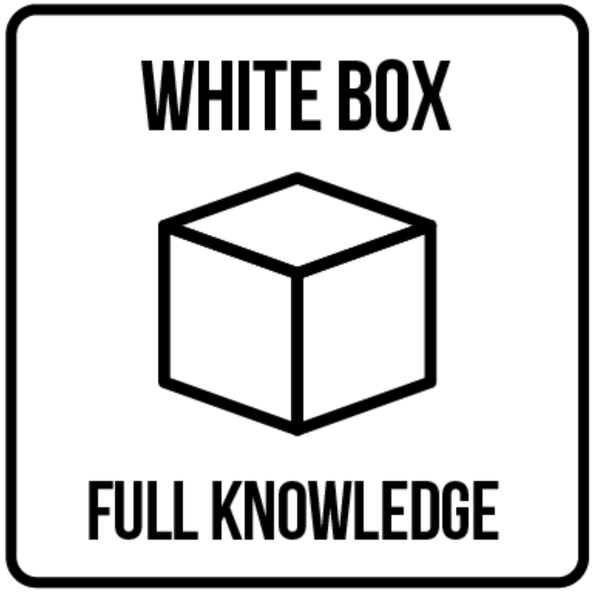
Phát hiện "điểm mù" của AI

Xu hướng AI Safety

Phương pháp tấn công

Các loại tấn công





TOG (Targeted Objectness Gradient): Phương thức tạo nhiễu đối kháng chuyên biệt nhằm qua mặt các bộ phát hiện đối tượng.


benign (No Attack)
TOG-untargeted
TOG-vanishing
TOG-fabrication
TOG-mislabeling
book: 0.28
bottle: 0.38
person: 0.32
person: 0.47
cup: 0.20
person: 0.22
person: 0.20
tie: 0.20

Hàm Loss trong mô hình nhận diện vật thể

$$\text{Total Loss} = \lambda_{\text{box}} \cdot \text{Loss}_{\text{box}} + \lambda_{\text{cls}} \cdot \text{Loss}_{\text{cls}} + \lambda_{\text{obj}} \cdot \text{Loss}_{\text{obj}}$$

Sai số vị trí kích thước khung bao (Bounding box)

Sai số phân loại lớp vật thể (Class probability)

Sai số độ tin cậy có vật thể hay không (Objectness score)

Phương pháp tối ưu hàm Loss:
Gradient Descent/Backpropagation

Phương pháp tấn công

Vanishing attack:

- Target: làm mất hết nhãn dán - *object_score của mọi ô = 0*

0	0	0	...	0	0
0	0	0	...	0	0
0	0	0	...	0	0
...	...	...	...	...	...
0	0	0	...	0	0
0	0	0	...	0	0

Fabrication attack:

- Target: làm xuất hiện vô số nhãn dán - *object_score của mọi ô = 1*

1	1	1	...	1	1
1	1	1	...	1	1
1	1	1	...	1	1
...	...	...	...	...	...
1	1	1	...	1	1
1	1	1	...	1	1

Hàm loss được dùng để đánh giá: objectness_loss

Phương pháp tấn công

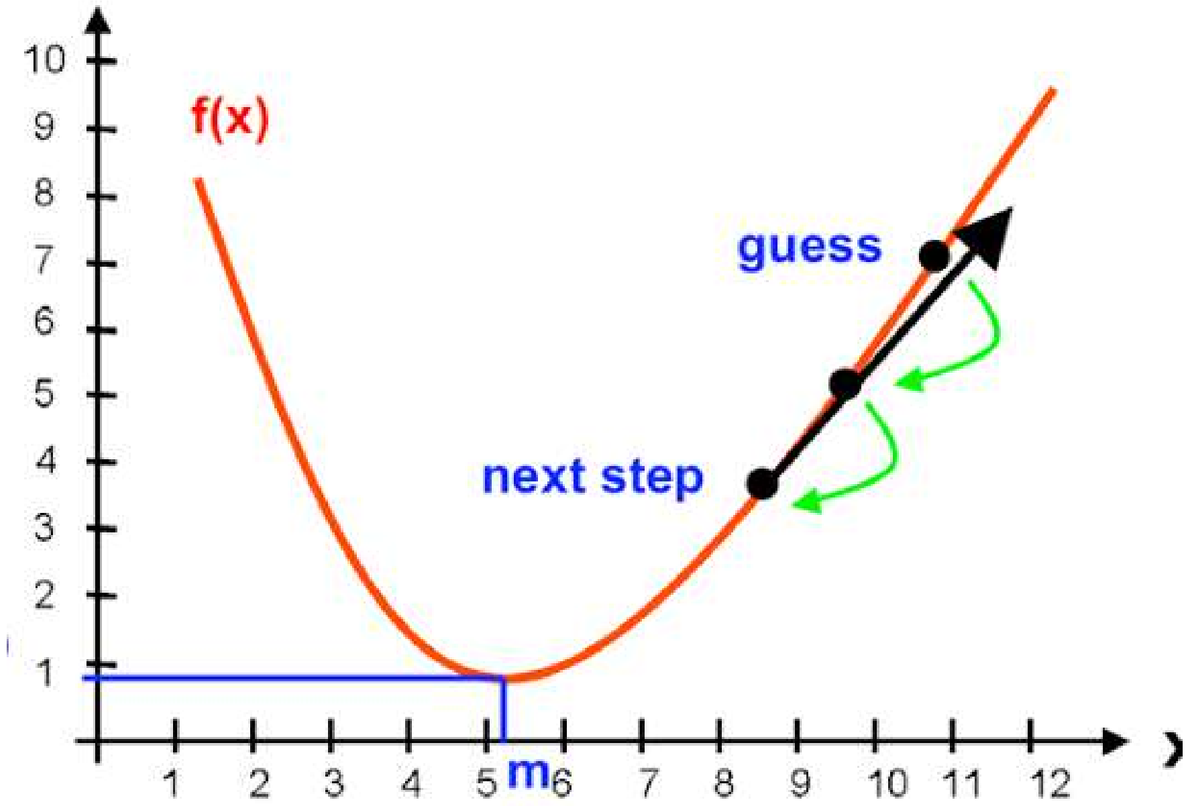
Mislabeling attack:

- Mục tiêu cốt lõi: Thay đổi nhãn phân loại của vật thể, nhưng khóa chặt vị trí khung giới hạn (Bounding Box).
- Bản chất: Tấn công có chủ đích (Targeted Attack).
- Thách thức thuật toán: Phải đánh lừa lớp phân loại (Classifier) mà không làm hỏng lớp định vị (Regressor) của mô hình nhận diện vật thể.

Phương pháp tấn công

Untargeted attack:

- Mục tiêu cốt lõi: Phá hỏng hoàn toàn kết quả nhận diện ban đầu bằng bất cứ giá nào.
- Kết quả chấp nhận được: Vật thể bị sai vị trí, biến mất, hoặc biến thành nhãn khác (Bất định).
- Bản chất: Tấn công phá hoại diện rộng, không cần đích đến cụ thể.



Chuẩn bị nghiên cứu

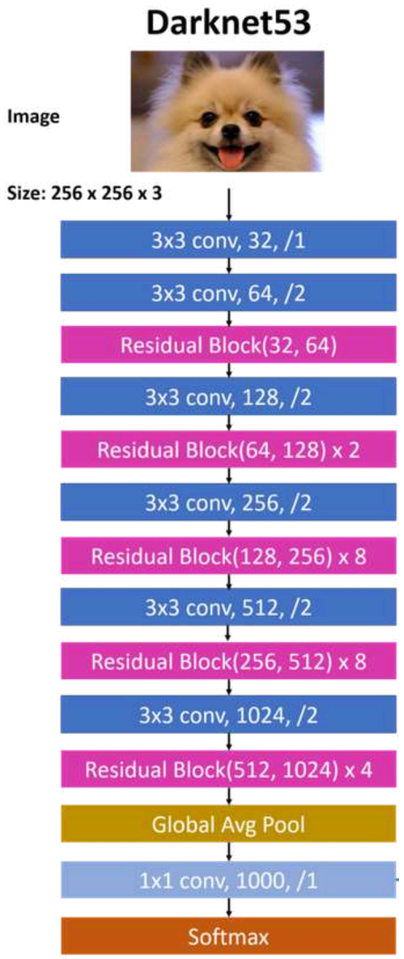
1. Lựa chọn mô hình nạn nhân
2. Lựa chọn bộ dữ liệu đánh giá (dataset)
3. Các chỉ số đánh giá (metrics)

Chuẩn bị nghiên cứu

Lựa chọn mô hình: YoloV3

YOLO (You Only Look Once) là một thuật toán/mô hình học sâu phổ biến được sử dụng rộng rãi trong bài toán nhận diện đối tượng (Object Detection), giúp máy tính trả lời đồng thời 2 câu hỏi cốt lõi trong bức ảnh

Chuẩn bị nghiên cứu



Mạng Darknet-53

Sử dụng các lớp Convolution 1 x 1 và 3 x 3 kết hợp Residual Blocks giúp trích xuất đặc trưng sâu mà không bị mất thông tin.

Phát hiện đa tỷ lệ (Multi-scale)

Dự đoán đối tượng ở 3 kích thước lưới khác nhau
(13 x 13, 26 x 26, 52 x 52) giúp nhận diện tốt cả vật thể nhỏ.

Chuẩn bị nghiên cứu

Dataset

- Tập dữ liệu gốc: **MS COCO 2017** (Validation Set)
- Benchmark tiêu chuẩn quốc tế cho bài toán Object Detection (5,000 ảnh, 80 lớp vật thể).
- Tập dữ liệu thực nghiệm: Chọn 100 ảnh ngẫu nhiên (2% tập Val).

Chuẩn bị nghiên cứu

Metrics

mAP (Mean Average Precision) : được dùng để **đánh giá hiệu năng tổng thể** của các **mô hình (YOLO)**. **mAP càng cao**, mô hình YOLO nhận diện **càng chuẩn xác.**

Chuẩn bị nghiên cứu

Metrics

Công thức :

$$mAP = \frac{1}{k} \sum_{i}^{k} AP_i$$

Trong đó :

N : tổng số class (ví dụ : class mèo, chó, ...)
APi : Điểm hiệu năng của riêng của class i.

Chuẩn bị nghiên cứu

Metrics

ASR (Attack Success Rate) : Tỷ lệ tấn công thành công. **ASR** đo lường **phần trăm số object bị làm nhiễu** thành công tại mô hình **YOLO**. **ASR càng cao**, **thuật toán tấn công YOLO càng hiệu quả.**

Chuẩn bị nghiên cứu

Metrics

Công thức :

$$ASR = \frac{N_{success}}{N_{total}} \times 100\%$$

Trong đó :

- **N total :** Tổng số đối tượng được đưa vào thử nghiệm tấn công.
- **N success :** Số đối tượng mà đầu ra của YOLO thỏa mãn chính xác tiêu chuẩn phá hoại của kẻ tấn công.

Chuẩn bị nghiên cứu

Metrics

PSNR (Peak Signal-to-Noise Ratio) : Tỷ số tín hiệu cực đại trên nhiễu. Dùng để **đánh giá** chất lượng **hình ảnh sau khi** thêm **nhiễu tấn công** vào **ảnh ban đầu**. **PSNR càng cao** chứng tỏ **ảnh bị biến đổi càng ít** (**càng khó nhận biết bằng mắt thường**).

Công thức :

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right)$$

Chuẩn bị nghiên cứu

Metrics

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right)$$ **(dB)**

Trong đó :

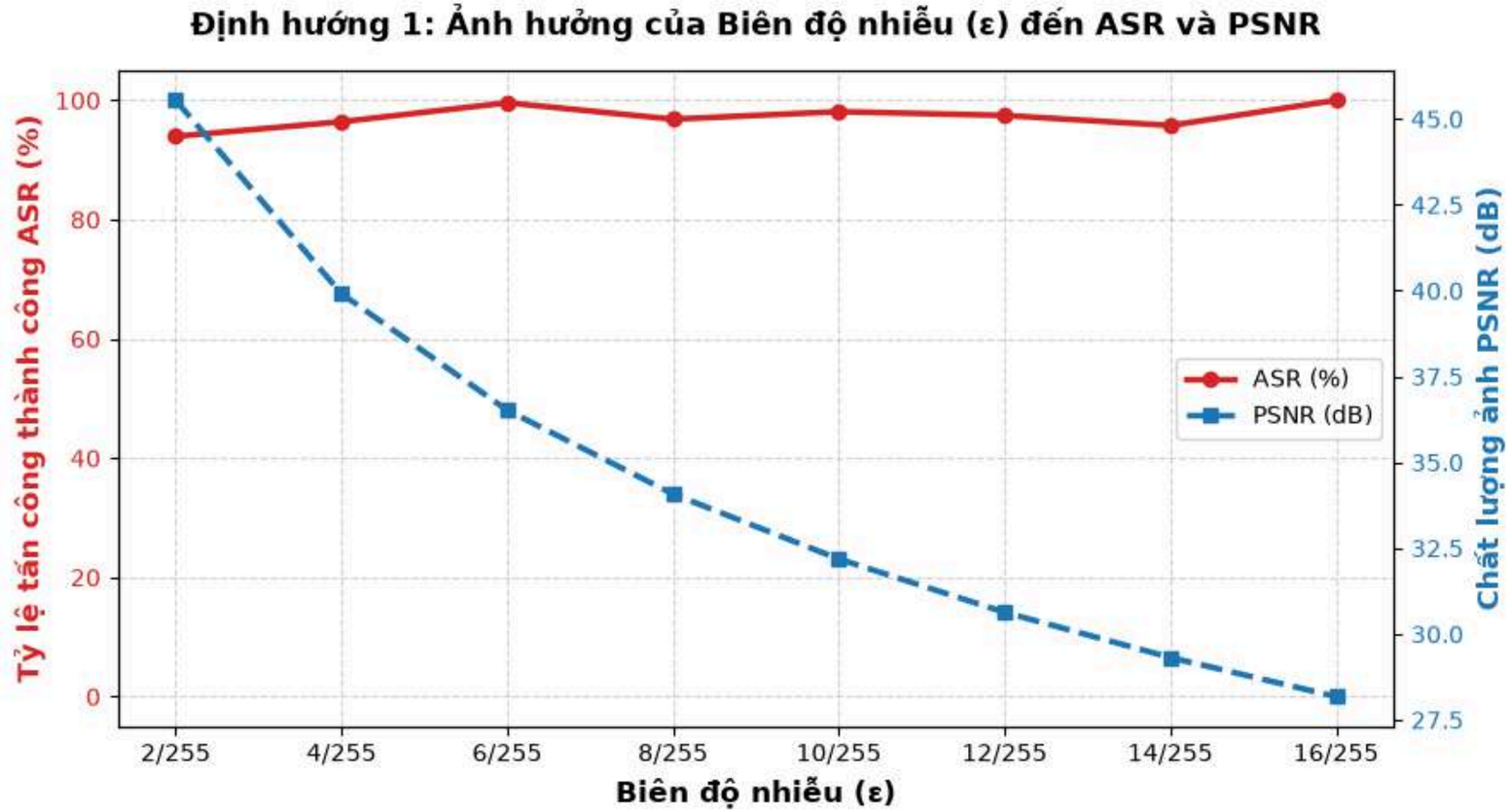
MAX_I : **Giá trị pixel tối đa** của ảnh, với ảnh màu 8-bit thông thường $MAX_I = 255$

$$MSE = \frac{1}{m \cdot n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2$$

Mean Squared Error : **Sai số toàn phương trung bình** giữa ảnh gốc **I** và ảnh đã bị biến đổi **K** có kích thước **m x n.**

Kết quả nghiên cứu

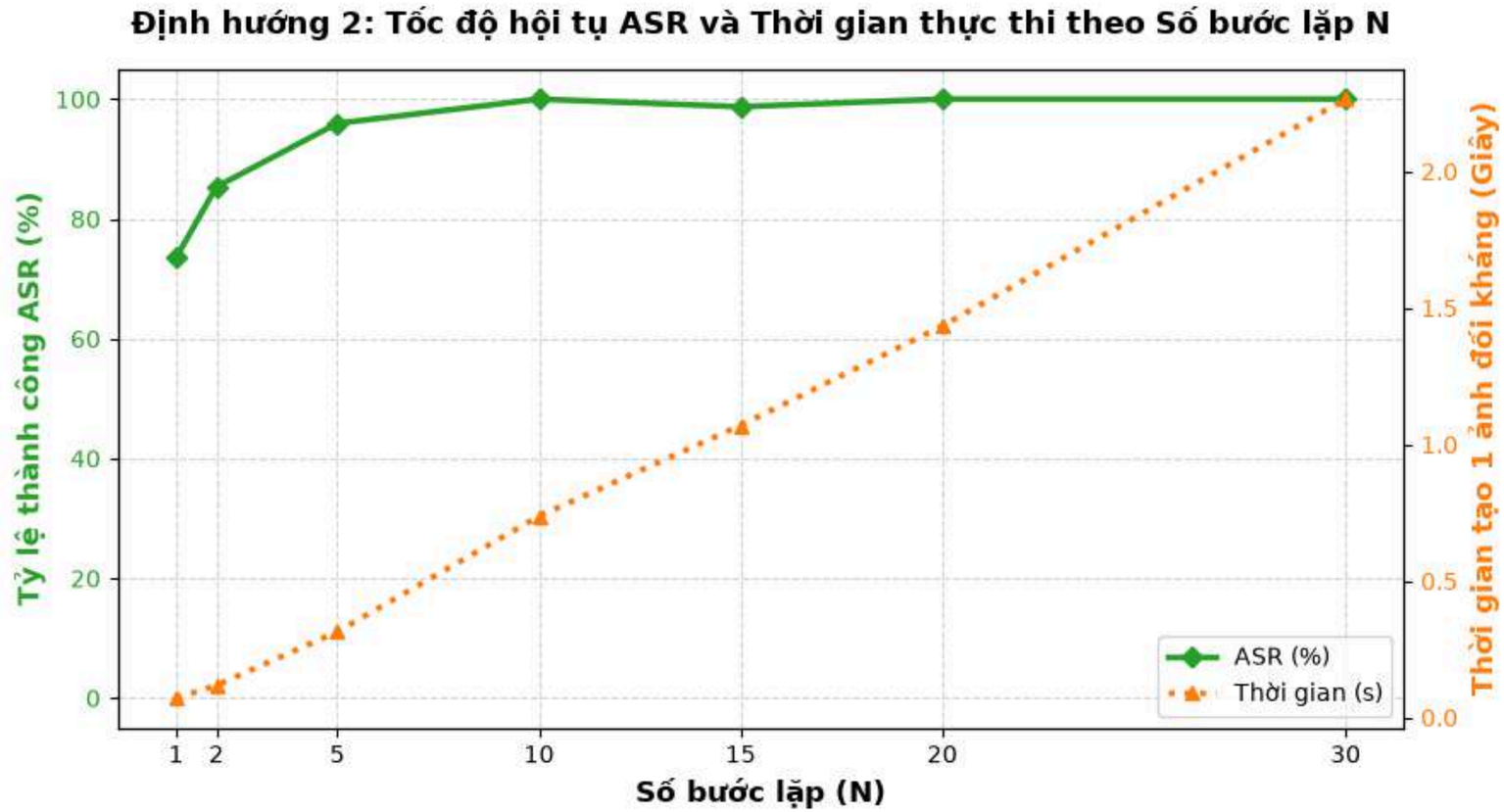
Biên độ nhiễu ε càng lớn, tỉ lệ tấn công thành công (ASR) càng cao nhưng chất lượng ảnh (PSNR) càng giảm.



Kết quả: Điểm cân bằng (Trade-off) là ngưỡng biên độ nhiễu ε mà ở đó ASR đủ cao (≈ 95%) nhưng PSNR vẫn giữ được mức chất lượng ảnh chấp nhận được → ε = 8/255

Kết quả nghiên cứu

Tăng số bước lặp N giúp tăng ASR nhưng tốn thời gian tính toán cho mỗi frame.

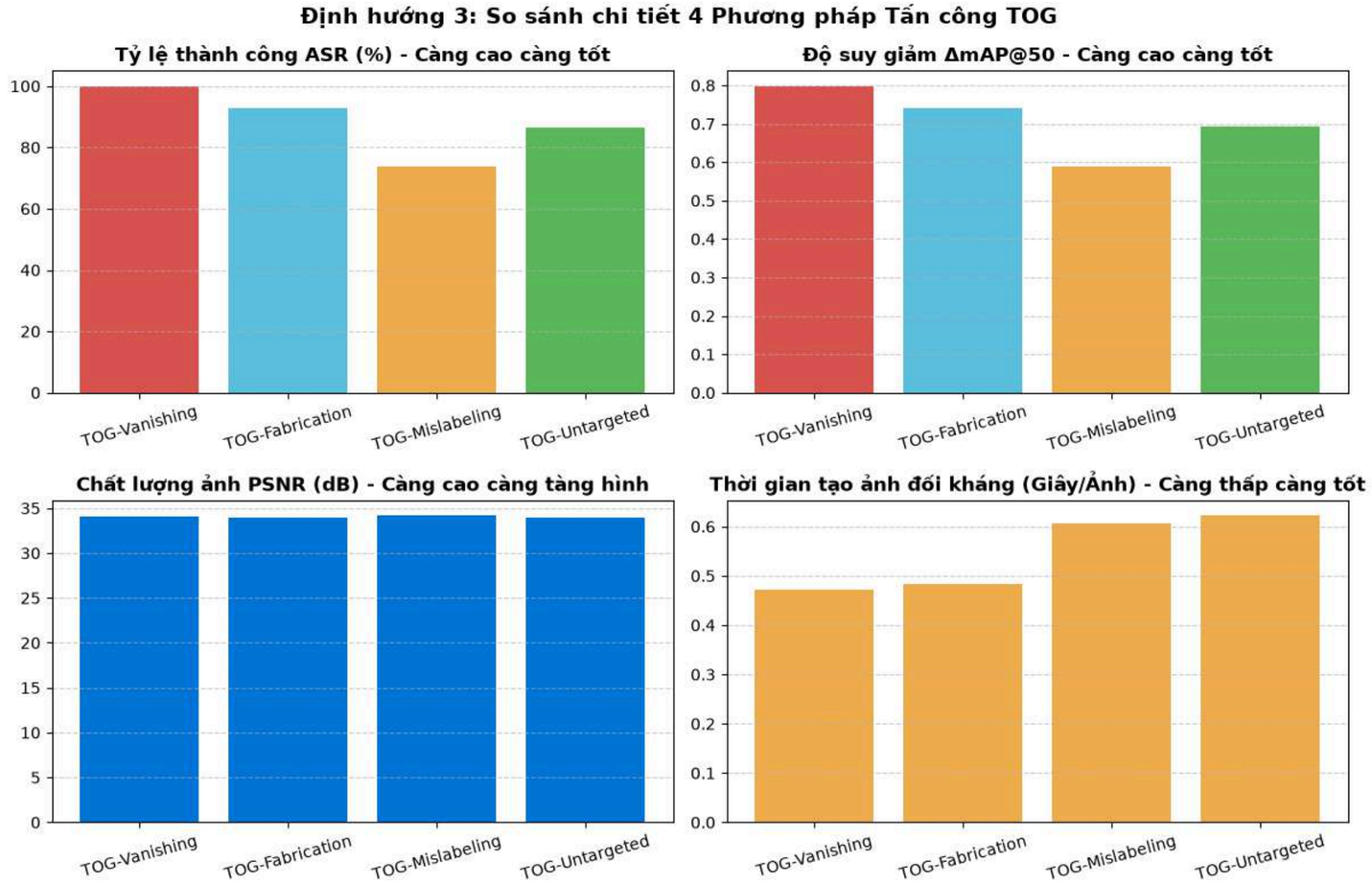


->Kết quả: Số bước lặp N tối ưu để vừa đạt ASR cao vừa tiết kiệm thời gian tính toán (tránh lặp thừa) → N = 10

Kết quả nghiên cứu

So sánh So kè 4 Phương pháp TOG

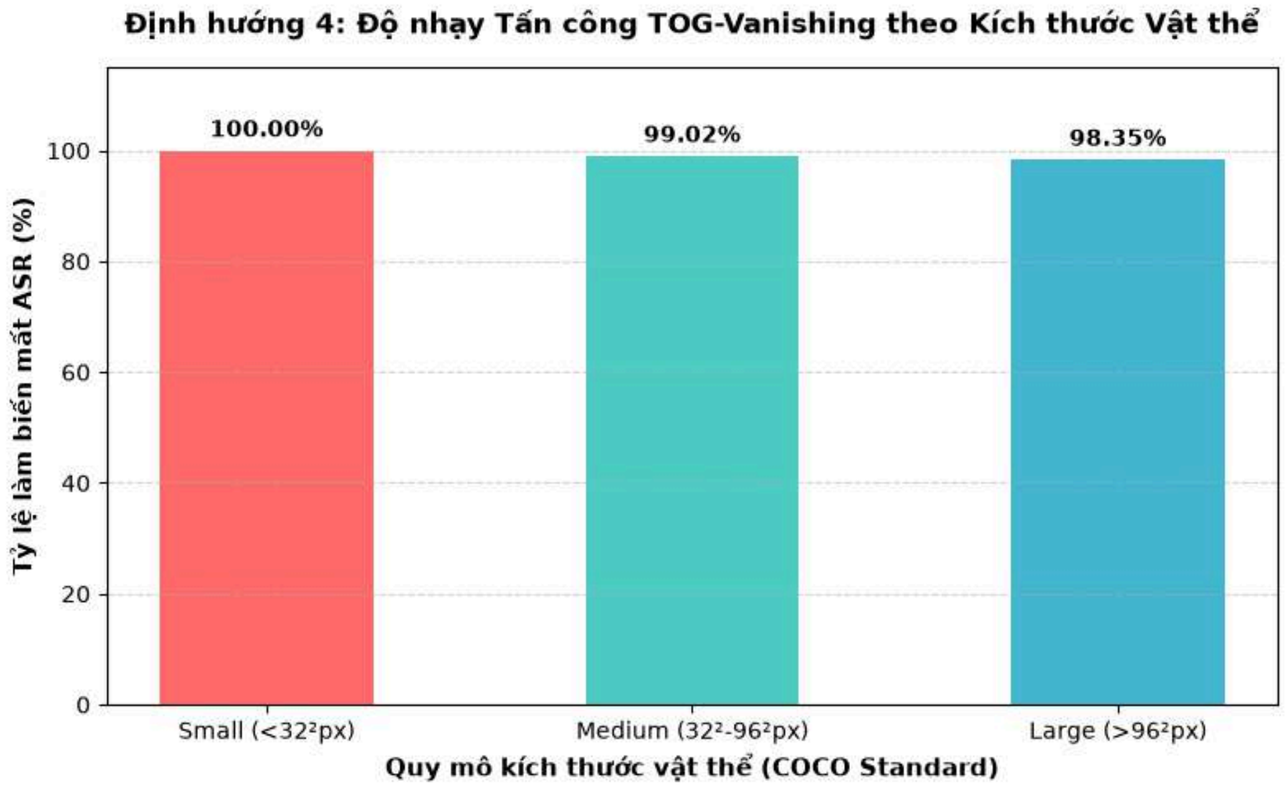
So sánh tog_vanishing, tog_fabrication, tog_mislabeling, và tog_untargeted trong cùng điều kiện ε = 8/255, N = 10.



Kết quả: Trong 4 phương pháp TOG, TOG-Vanishing đạt hiệu quả phá hoại mạnh nhất (ASR ~100%, ΔmAP@50 ~0.8) và tối ưu thời gian nhất (~0.47s), trong khi tất cả phương pháp đều duy trì khả năng tàng hình thị giác đồng nhất (PSNR ~34 dB).

Kết quả nghiên cứu

Độ nhạy tấn công theo Kích thước vật thể (Object Size)
Đánh giá xem nhóm vật thể Small (<32^2 px), Medium, hay Large (>96^2 px) bị biến mất dễ nhất dưới tác động của TOG-vanishing.

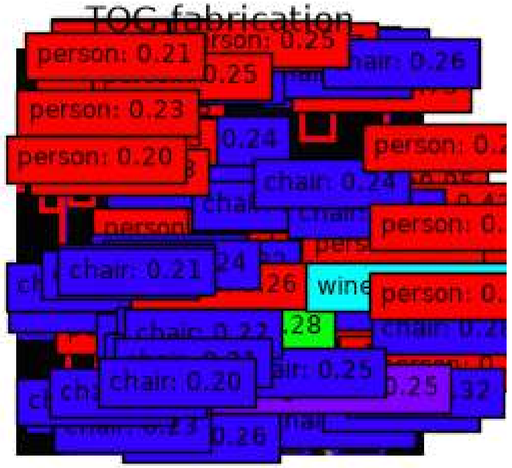


Kết quả: Tấn công TOG-Vanishing nguy hiểm trên mọi quy mô kích thước, trong đó các đối tượng nhỏ (như biển báo giao thông xa, người đi bộ xa) là nhóm dễ bị biến mất nhất.

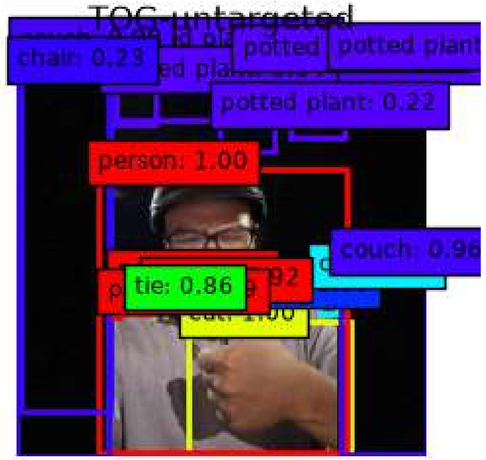
Kết quả nghiên cứu

TOG-vanishing

TOG-fabrication

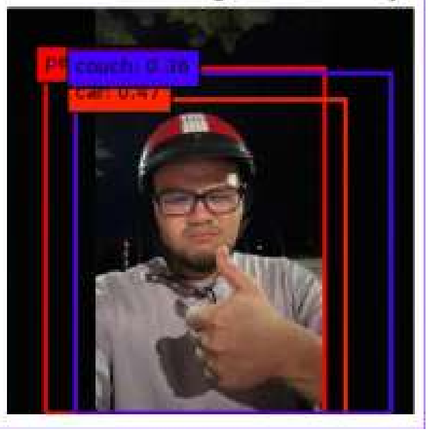


TOG-untargeted



TOG-mislabeling (Most-Likely)

TOG-mislabeling (Least-Likely)



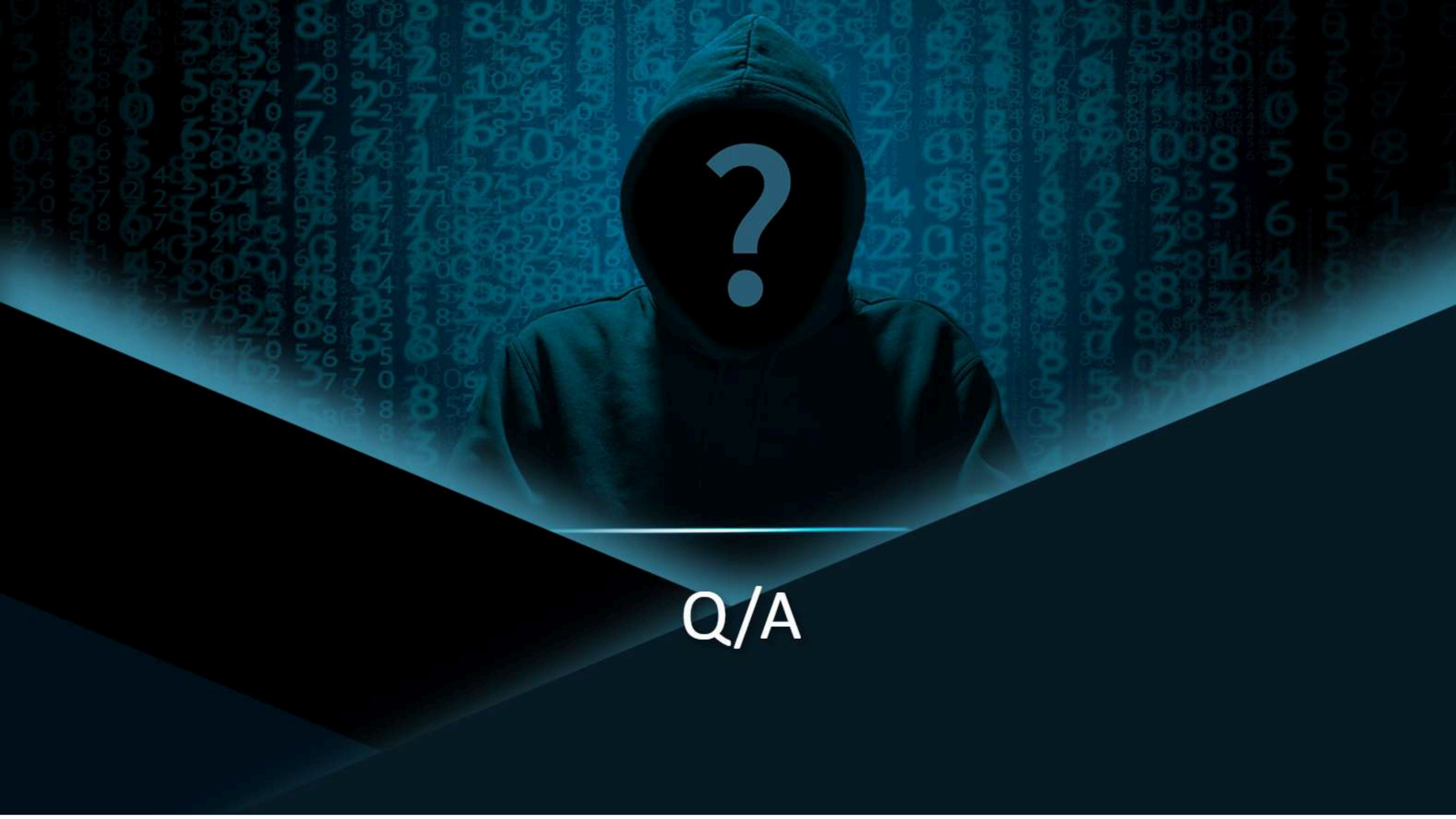
Giải pháp phòng thủ & Khắc phục điểm yếu

1. Huấn luyện đối kháng
(Adversarial Training)

2. Tiền xử lý & Lọc nhiễu đầu vào
(Input Preprocessing)

3. Trích xuất đặc trưng kháng nhiễu
(Feature Denoising)

Thank you


Q/A

=> đặc biệt nguy hiểm với xe tự lái: các biển báo giao thông ở xa hoặc người đi bộ ở khoảng cách xa (hiển thị kích thước nhỏ trên camera) sẽ là những đối tượng bị biến mất đầu tiên

4 loại phương pháp tấn công

Vanishing attack: làm mất hết nhãn dán
target của mô hình này sẽ quy về mô hình gồm mọi ô đều có object score = 0
→ khi mọi ô tiến tới 0 (và bé hơn ngưỡng threshold) thì object score có lẽ sẽ được gán = 0 thì bounding box của ô đấy sẽ bị mất đi

Fabrication attack: Tạo thêm nhiều nhãn nhiễu để mô hình NMS (non-maximum suppression) bị quá tải

vậy NMS là gì?

target của mô hình này sẽ quy về mô hình gồm mọi ô đều có object score = 1

→ khi mọi ô tiến tới 1 (và vượt qua ngưỡng threshold) thì sẽ có giá trị nhãn mới xuất hiện, vậy với mỗi ô bounding box thì nhãn được lấy từ đâu ra?

tại sao nó đạt hiệu quả cao


benign (No Attack)
TOG-untargeted
TOG-vanishing
TOG-fabrication
TOG-mislabeling
benign (No Attack)
TOG-untargeted
TOG-vanishing
TOG-fabrication
TOG-mislabeling
benign (No Attack)
TOG-untargeted
TOG-vanishing
TOG-fabrication
TOG-mislabeling

4 loại phương pháp tấn công

GIỐNG NHAU:

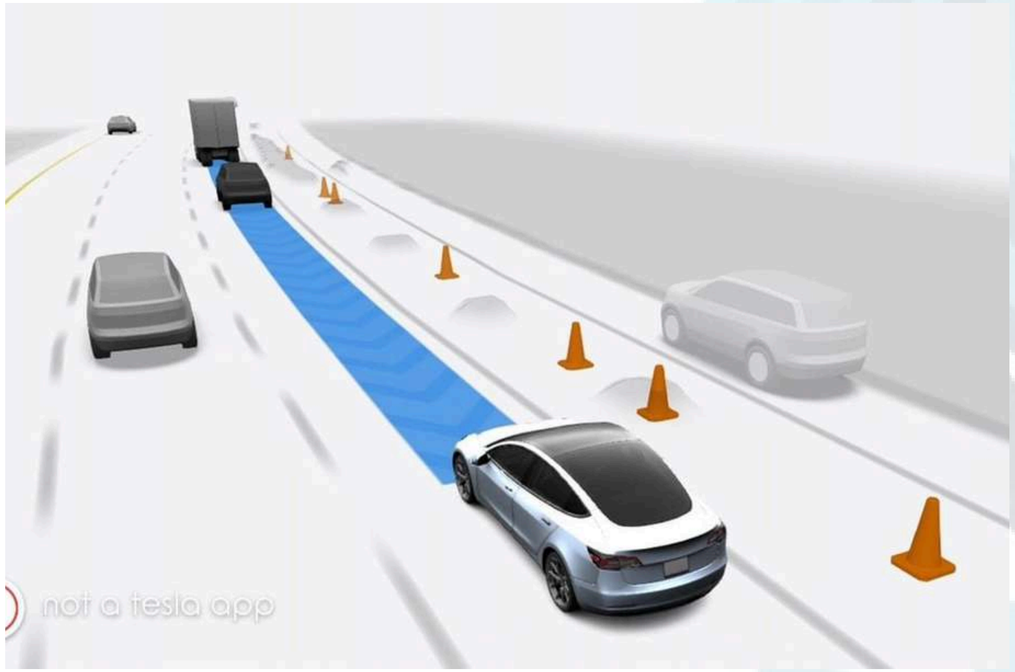
Điểm chung: **dùng phương pháp gradient descent để tối ưu hàm Loss**
Hàm Loss tập trung vào việc tối ưu object_score giữa (target_image) và (adversarial_image)

bounding box: gồm X, Y, W, H
X, Y: tâm của center và nó thường nằm trong vật
W: width, H: height (kích thước của bounding box) ? Vậy bounding box là gì? Khung nào?

GIải thích sâu hơn về quá trình nhận diện:
0. isValidate value, ảnh có đủ các thông số;
   1. Khởi tạo ảnh nhiễu
   2. Tấn công:
      a. gradient = compute_object_...(adv_img)

đi sâu hơn vào hàm compute;
aa: tạo ra target với full object_score = 0;
ab: thông qua lan truyền ngược, tính toán gradient cho từng pixel: trong này có đạo hàm loss / đạo hàm gradient
ac: trả về hàm gradient thỏa mãn;'
      b. adv_img = adv_img - np.sign(gradient) * step_size // tại sao * step_size
biết là hàm đi xuống như mà cơ chế hoạt động như này là sao

Vụ tai nạn của Joshua Brown năm 2016



2. Giới thiệu mô hình YOLOv3

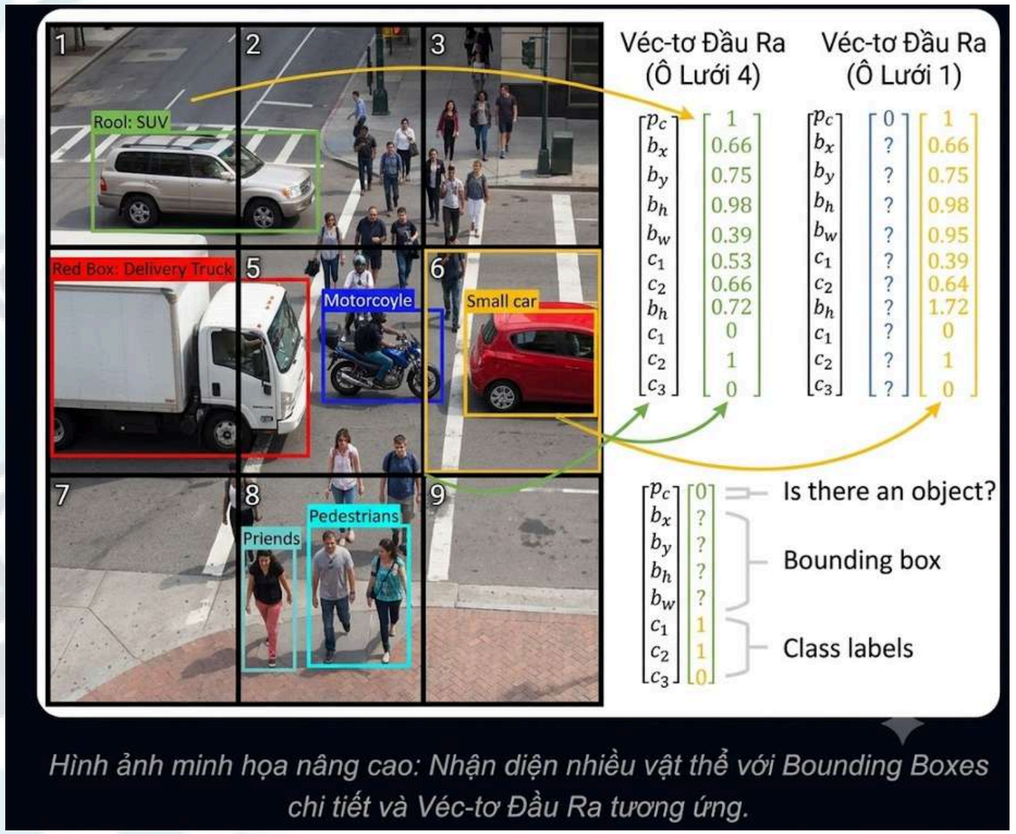
Định nghĩa:

YOLO(You Only Look Once) là một thuật toán/mô hình học sâu phổ biến được sử dụng rộng rãi trong bài toán nhận diện đối tượng (Object Detection), giúp máy tính trả lời đồng thời 2 câu hỏi cốt lõi trong bức ảnh:

- Có cái gì?(Phân loại đối tượng - Class)
- Nằm ở đâu?(Xác định tọa độ - Bounding Box

Cơ chế One-pass (Real-time):

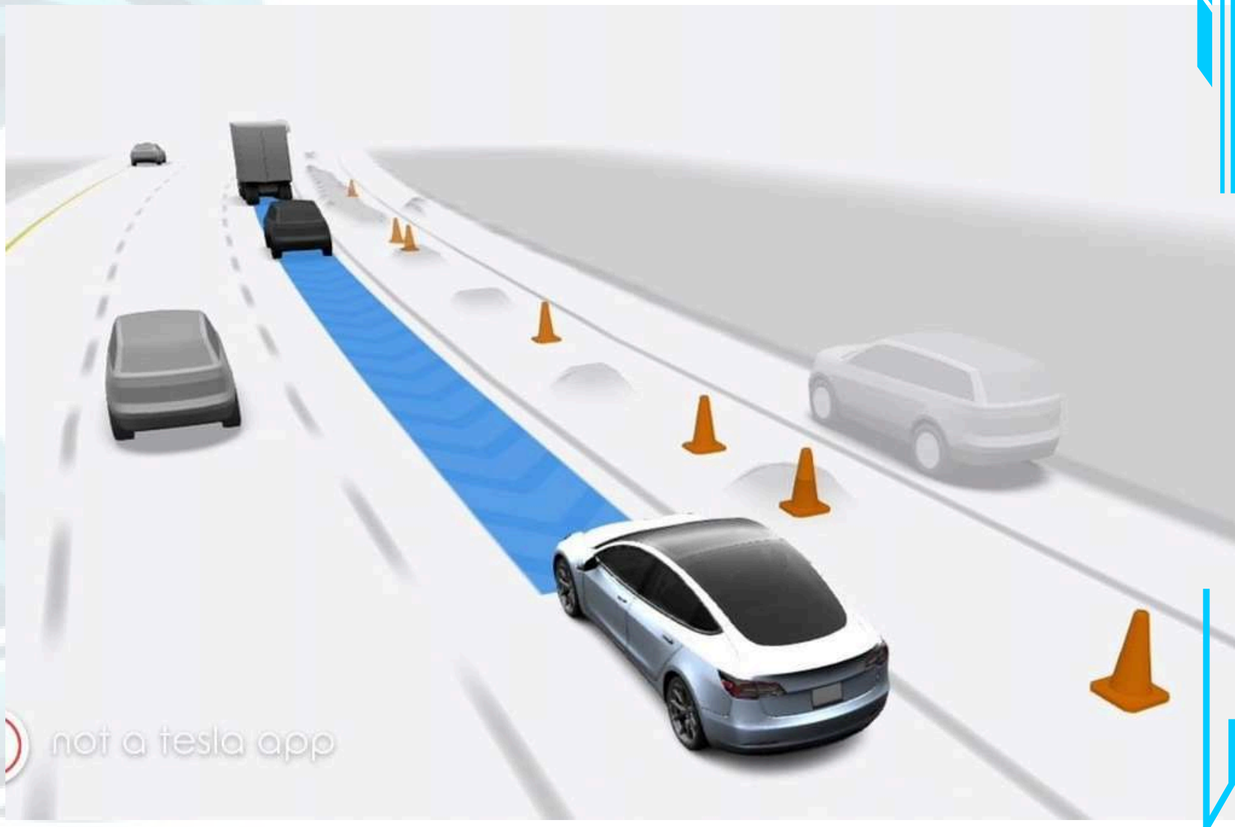
- Dự đoán Bounding Box và Class cùng một lúc chỉ qua 1 lần duyệt ảnh.



Hình ảnh minh họa nâng cao: Nhận diện nhiều vật thể với Bounding Boxes chi tiết và Véc-tơ Đầu Ra tương ứng.

Vụ tai nạn của Joshua Brown năm 2016 là minh chứng rõ nhất về việc nhiễu thị giác (vision noise) trên camera khiến AI nhận diện sai và gây chết người.

- Joshua Brown bật chế độ tự lái (Autopilot) trên chiếc Tesla Model S chạy trên đường cao tốc tại Florida. Một chiếc xe tải lớn màu trắng bất ngờ rẽ ngang qua lộ trình của Tesla.
- Thân xe tải có màu trắng sáng, trùng hợp với bầu trời phía sau lúc đó đang bị nắng gắt chiếu lóa (cháy sáng).
- Hệ thống camera thuần túy (pure vision) của Tesla bị nhiễu do độ tương phản bằng không. AI hoàn toàn không nhận diện được chiếc xe tải mà lầm tưởng đó là bầu trời trống trải hoặc một biển báo trên cao.
- Xe Tesla giữ nguyên tốc độ, đâm thẳng vào gầm xe tải khiến phần mui xe bị xé toạc và người lái tử vong tại chỗ.



2. Giới thiệu mô hình YOLOv3

Cách mô hình YOLOv3 được train:

Mục tiêu: Cập nhật các trọng số (weights) của mô hình để giảm thiểu sai số giữa dự đoán và thực tế

Quá trình huấn luyện mô hình YOLOv3

- Khởi tạo trọng số (*W*)
- Chia lưới
- Dự đoán đa thành phần: Dựa trên 3 chỉ số
- Tối ưu hóa bằng Gradient Descent:
  - Hàm mất mát (*L*)
  - Cập nhật trọng số

Kết quả: Quá trình này giúp giảm thiểu sai số, giúp xác định chính xác tọa độ tâm và kích thước của bounding box.

1. Lí do thực hiện dự án:

1.1 Đảm bảo An toàn cho các hệ thống thực tế:

Hầu hết các hệ thống tự động hiện đại phụ thuộc vào Object Detection để đưa ra quyết định trong thời gian thực:
VD: Xe tự lái, CCTV, y tế.

1.2 Phát hiện "điểm mù" của AI: Đánh giá rủi ro mà các phương pháp kiểm thử truyền thống bỏ sót.

1.3 Xu hướng AI Safety: Đóng góp vào lĩnh vực bảo mật AI — ưu tiên hàng đầu của các tập đoàn công nghệ lớn.

4 loại phương pháp tấn công

Thuật toán tấn công

Mislabeling attack:

- Giữ nguyên không gian: Kế thừa tọa độ thật (x, y, w, h) và điểm tin cậy (Objectness = 1.0).
- Chỉnh sửa One-Hot Vector:
  - Ép giá trị nhãn thật về 0.0.
  - Bơm giá trị nhãn giả mạo lên 1.0.
- Chiến lược chọn nhãn giả:
  - most_likely: Nhãn sai có xác suất cao thứ hai (Dễ đánh lừa nhất).
  - least_likely: Nhãn sai có xác suất thấp nhất (Khó nhất, sai lệch ngữ nghĩa lớn nhất).

Untargeted attack:

- Mục tiêu ngắm bắn: Chính là kết quả đúng ban đầu (Ground Truth).
- Thủ thuật đảo ngược: Thêm dấu âm (-) vào trước hàm mất mát.
- Công thức: Loss_untargeted = - Loss(ŷ, y)
- Nguyên lý: Cực tiểu hóa một hàm số âm tương đương với việc cực đại hóa độ sai lệch của hàm số đó.

SEAS