INCHEON NATIONAL UNIVERSITY
COMPUTER VISION LABORATORY

# Adversarial Attention Perturbations for Large Object Detection Transformers
# (ICCV 2025)

**Undergraduate Researcher at CVLab**
**Lee Dohyeong**

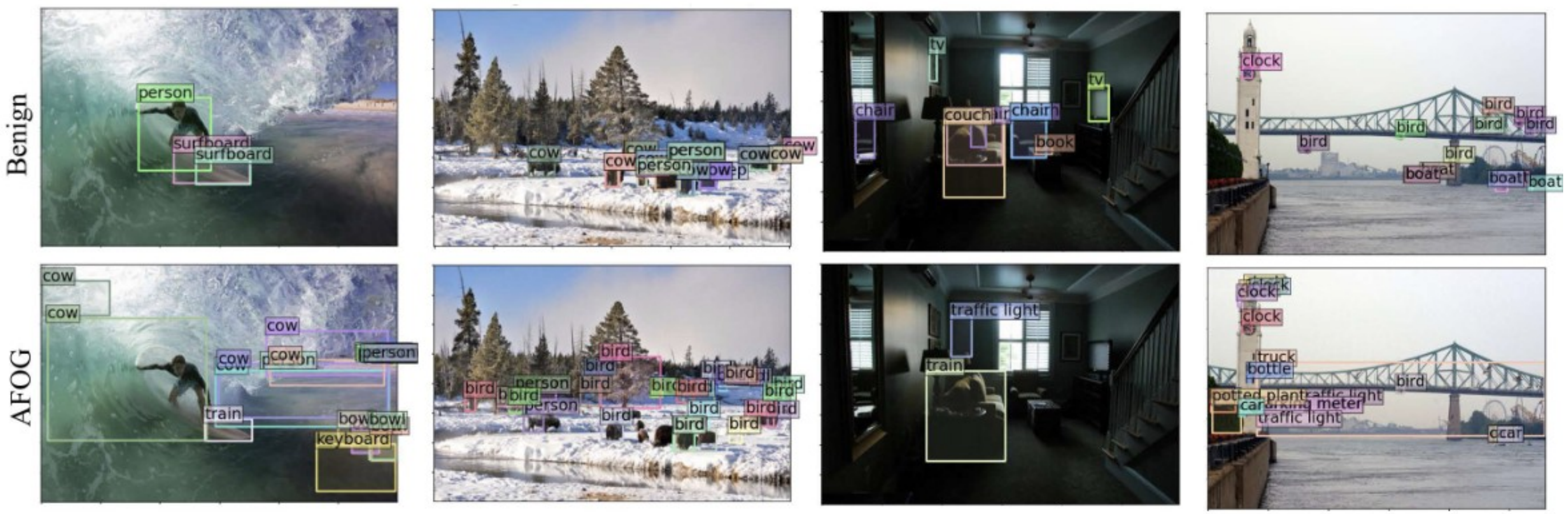
**2025.9.12**

# Contents


- Introduction
- Related Work
- Method
- Experiments
- Conclusion

# Introduction

- **Adversarial Attack** :
  - by adding tiny, human-imperceptible noise (perturbations) to the original data.
  - Benign Image -> Adv Attack(Noise) -> Adv Image



# Introduction

- **Existing attack methods limitations**

  1. Designed for CNN-based detectors and are less effective against Transformer models.
  2. Transformer-specific attacks cannot be applied to CNN models.

=> architecture-agnostic framework is needed to attack both effectively.

# Introduction

- **Contribution :**

Proposes a new attack method called AFOG (Attention-Focused Offensive Gradient).

1. **Neural-Architecture Agnostic Framework**
2. **Learnable Attention Mechanism**
3. **Integrated Attack Loss**
4. **Efficiency and Stealth**(generates visually imperceptible perturbations rapidly)

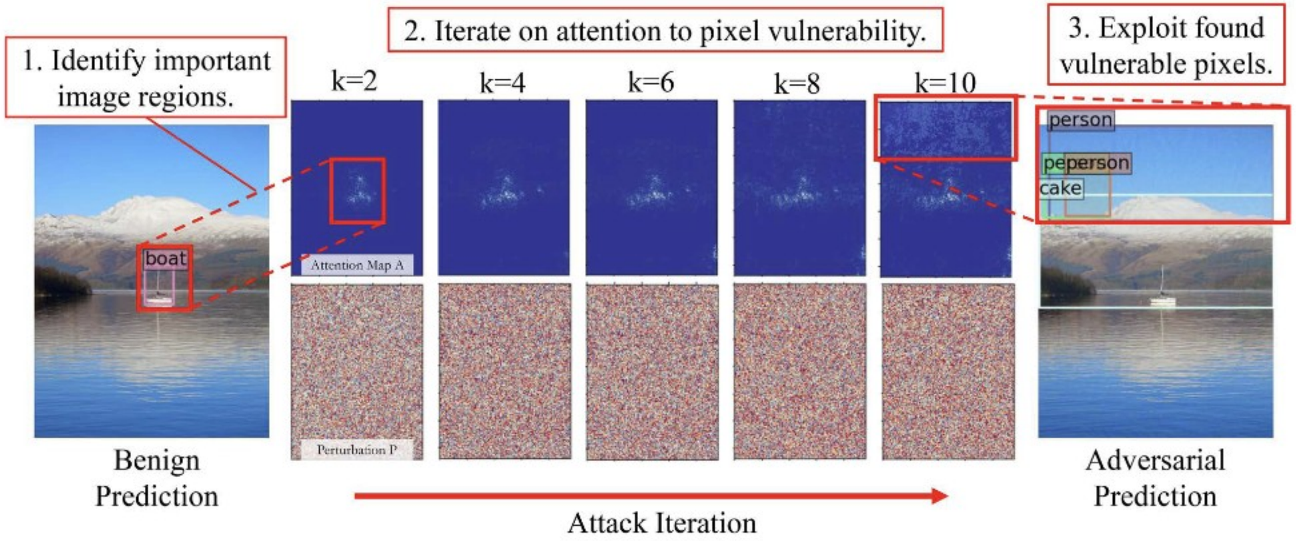
=> experiments on twelve state-of-the-art detection transformers.

# Related Work

- Categorization of Adversarial Attacks
  - Black Box Attack(Surrogate-based) ; (UEA, RAD, and GHFD.)
    - The attacker has no access to the internal information of the victim model.
    - They generate attacks on a substitute (surrogate) model and then transfer them
    - **Limitation:** Their performance significantly drops when the surrogate and victim models have different architectures.
  - White Box Attack(Victim-based) ; (EBAD, OATB, and AttentionFool)
    - The attacker has full access to the victim model's internal information, such as its architecture, parameters, and loss function.
    - **Limitation:** Existing methods are often architecture-specific (e.g., only for Transformers) or show inconsistent performance.

# Method

- **AFOG Attack**
  - **'Perturbation P'**, is the random noise we use as our tool to disrupt the image.
  - **'Attention Map A'**, is the guide. It learns where to focus the noise for maximum effect.
  - **the attack iterates** (k=2 to k=10), the Attention Map finds the most vulnerable pixels,



# Method

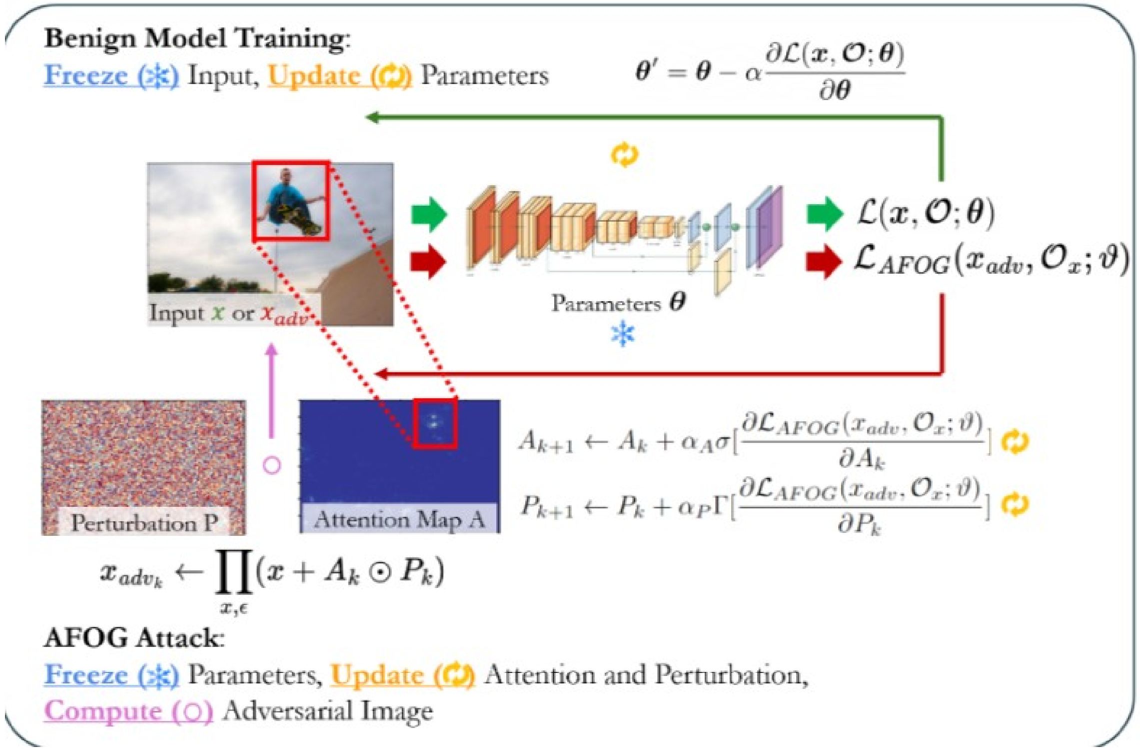
- **Victim Detector** $f_D(\vartheta, x)$
  - x : 탐지되어야 할 Nx의 객체
  - O__i: 탐지기 F__D의 인식 대상 $\mathcal{O}_x = \{O_1, O_2, \ldots, O_{N_x}\}$
  - 정상적인 예측 : $R[f_D(\vartheta, x)] = \{(b_i, c_i) | i = 1, \ldots, N_x\}$
    - b__i : Bounding box 예측 결과
    - c__i : cls lable 예측 결과
    - 기준 : (B__i, C__i) IOU 〉= 0.5

# Method

- **AFOG Attack Goal**
  - **Attention 기반 반복적 학습 메커니즘 ->X(원본이미지)에 적대적 교란 P추가 -> X_adv(적대적 이미지)생성**
  - **Goal :** Find an adversarial example $x_{adv}$ that maximizes the success rate of mis-detection for all images in dataset $D$.
  - **왜곡 제약** : X_adv는 $min||x - x_{adv}||_p$ 만족해야 한다.
    - 적대적 교란이 원본 이미지와 시각적으로 거의 구별할 수 없게 해야하기 때문

# Method

- **AFOG Attack Algorithm**



# Method

- **AFOG Attack Algorithm**
  - **Initialize**

**Algorithm 1** AFOG attack on an input image.

**Require:** Victim image $x \in \mathcal{D}$, test-set $\mathcal{D}$, Victim pre-trained model $f_D(\vartheta)$, Perturbation step size $\alpha_P$, Attention step size $\alpha_A$, Number of iterations $T$, Maximum perturbation $\epsilon$.

Initialize $\mathcal{O}_x \leftarrow f_D(x;\vartheta)$
Initialize attention map $A_0 \leftarrow 1$;
Initialize perturbation $P_0 \leftarrow Random(-\epsilon, \epsilon)$;
Initialize step variable $k \leftarrow 1$;
**while** $k \leq T$ **do**
  Attack image $x_{adv\,k} \leftarrow \prod_{x,\epsilon}(x + A_k \odot P_k)$;
  Forward propagate $x_{adv}$ through $f_D(\vartheta, x_{adv})$;
  Compute bbox-loss $\mathcal{L}_{bbox}(x_{adv}, \mathcal{O}_x; \vartheta)$;
  Compute cls-loss $\mathcal{L}_{cls}(x_{adv}, \mathcal{O}_x; \vartheta)$;
  $\mathcal{L}_{AFOG}(x_{adv}, \mathcal{O}_x; \vartheta)$ = bbox-loss + cls-loss;
  Calculate losses with respect to $A_k$ and $P_k$: $\mathcal{L}_A(x_{adv}, \mathcal{O}_x; \vartheta), \mathcal{L}_P(x_{adv}, \mathcal{O}_x; \vartheta)$;
  Normalize attention loss $\mathcal{L}_A \leftarrow \mathrm{Norm}(\mathcal{L}_A)$;
  Take sign of perturbation loss $\mathcal{L}_P \leftarrow \mathrm{Sign}(\mathcal{L}_P)$;
  $A_{k+1} \leftarrow A_k - \alpha_A \mathcal{L}_A$;
  $P_{k+1} \leftarrow P_k - \alpha_P \mathcal{L}_P$;
  $k \leftarrow k + 1$;
**end while**
$x_{adv\,k+1} \leftarrow \prod_{x,\epsilon}(x + A_{k+1} \odot P_{k+1})$;
**return** $x_{adv}$

# Method

- **AFOG Attack Algorithm**

**Algorithm 1** AFOG attack on an input image.

**Require:** Victim image $x \in \mathcal{D}$, test-set $\mathcal{D}$, Victim pre-trained model $f_D(\vartheta)$, Perturbation step size $\alpha_P$, Attention step size $\alpha_A$, Number of iterations $T$, Maximum perturbation $\epsilon$.

Initialize $\mathcal{O}_x \leftarrow f_D(x;\vartheta)$
Initialize attention map $A_0 \leftarrow 1$;
Initialize perturbation $P_0 \leftarrow Random(-\epsilon,\epsilon)$;
Initialize step variable $k \leftarrow 1$;
**while** $k \leq T$ **do**
    Attack image $x_{adv\,k} \leftarrow \prod_{x,\epsilon}(x + A_k \odot P_k)$;
    Forward propagate $x_{adv}$ through $f_D(\vartheta, x_{adv})$;
    Compute bbox-loss $\mathcal{L}_{bbox}(x_{adv}, \mathcal{O}_x;\vartheta)$;
    Compute cls-loss $\mathcal{L}_{cls}(x_{adv}, \mathcal{O}_x;\vartheta)$;
    $\mathcal{L}_{AFOG}(x_{adv}, \mathcal{O}_x;\vartheta) =$ bbox-loss + cls-loss;
    Calculate losses with respect to $A_k$ and $P_k$: $\mathcal{L}_A(x_{adv}, \mathcal{O}_x;\vartheta), \mathcal{L}_P(x_{adv}, \mathcal{O}_x;\vartheta)$;
    Normalize attention loss $\mathcal{L}_A \leftarrow$ Norm($\mathcal{L}_A$);
    Take sign of perturbation loss $\mathcal{L}_P \leftarrow$ Sign($\mathcal{L}_P$);
    $A_{k+1} \leftarrow A_k - \alpha_A \mathcal{L}_A$;
    $P_{k+1} \leftarrow P_k - \alpha_P \mathcal{L}_P$;
    $k \leftarrow k + 1$;
**end while**
$x_{adv\,k+1} \leftarrow \prod_{x,\epsilon}(x + A_{k+1} \odot P_{k+1})$;
**return** $x_{adv}$

$$x_{adv_k} \leftarrow \prod_{x,\epsilon}(x + A_k \odot P_k) \quad (1)$$

$$\begin{aligned}\mathcal{L}_{AFOG}(x_{adv}, \mathcal{O}_x;\vartheta) &= \mathcal{L}_{bbox}(x_{adv}, \mathcal{O}_x;\vartheta) \\ &+ \mathcal{L}_{cls}(x_{adv}, \mathcal{O}_x;\vartheta)\end{aligned} \quad (3)$$

$$\mathcal{L}_{bbox}(x_{adv}, \mathcal{O}_x;\vartheta) = \sum_{i=1}^{N_x}[f_\vartheta(x, o_i) - f_\vartheta(x_{adv}, o_{adv_i})] \quad (4)$$

$$\mathcal{L}_{cls}(x_{adv}, \mathcal{O}_x;\vartheta) = \sum_{i=1}^{N_x}[f_\vartheta(x, c_i) - f_\vartheta(x_{adv}, c_{adv_i})] \quad (5)$$

# Method

- **AFOG Attack Algorithm**

**Algorithm 1** AFOG attack on an input image.

**Require:** Victim image $x \in \mathcal{D}$, test-set $\mathcal{D}$, Victim pre-trained model $f_D(\vartheta)$, Perturbation step size $\alpha_P$, Attention step size $\alpha_A$, Number of iterations $T$, Maximum perturbation $\epsilon$.

```
Initialize 𝒪_x ← f_D(x; ϑ)
Initialize attention map A_0 ← 1;
Initialize perturbation P_0 ← Random(−ϵ, ϵ);
Initialize step variable k ← 1;
while k ≤ T do
    Attack image x_advk ← ∏_{x,ϵ}(x + A_k ⊙ P_k);
    Forward propagate x_adv through f_D(ϑ, x_adv);
    Compute bbox-loss ℒ_bbox(x_adv, 𝒪_x; ϑ);
    Compute cls-loss ℒ_cls(x_adv, 𝒪_x; ϑ);
    ℒ_AFOG(x_adv, 𝒪_x; ϑ) = bbox-loss + cls-loss;
    Calculate losses with respect to A_k and P_k:
    ℒ_A(x_adv, 𝒪_x; ϑ), ℒ_P(x_adv, 𝒪_x; ϑ);
    Normalize attention loss ℒ_A ← Norm(ℒ_A);
    Take sign of perturbation loss ℒ_P ← Sign(ℒ_P);
    A_{k+1} ← A_k − α_A ℒ_A;
    P_{k+1} ← P_k − α_P ℒ_P;
    k ← k + 1;
end while
x_advk+1 ← ∏_{x,ϵ}(x + A_{k+1} ⊙ P_{k+1});
return x_adv
```

$$A_{k+1} \leftarrow A_k + \alpha_A \sigma[\frac{\partial \mathcal{L}_{AFOG}(x_{adv}, \mathcal{O}_x; \vartheta)}{\partial A_k}] \quad (6)$$

$$P_{k+1} \leftarrow P_k + \alpha_P \Gamma[\frac{\partial \mathcal{L}_{AFOG}(x_{adv}, \mathcal{O}_x; \vartheta)}{\partial P_k}] \quad (7)$$

# Method

- **Special Cases of the AFOG Attack**
  - **AFOG – V(객체 소멸 공격)**
    - **Goal:** Ensure that no objects are detected; all detected objects should be eliminated.
    - **Method:**
      - The difference lies in the initialization state.
      - The empty set assumed as the ground truth.

$$\mathcal{L}_{AFOG_V}(x_{adv}, \mathcal{O}_x; \vartheta) = -\mathcal{L}_{bbox}(x_{adv}, \varnothing; \vartheta) - \mathcal{L}_{cls}(x_{adv}, \varnothing; \vartheta) \tag{8}$$

  - **AFOG – F(오탐지 공격)**
    - **Goal:** making the model detect objects that do not actually exist.
    - **Method:**
      - Remove the original IoU thre … ons.

$$\mathcal{L}_{AFOG_F}(x_{adv}, \mathcal{O}_x; \vartheta) = -\mathcal{L}_{bbox}(x_{adv}, \mathcal{O}_F; \vartheta) - \mathcal{L}_{cls}(x_{adv}, \mathcal{O}_F; \vartheta) \tag{9}$$

# Experiments

Table 1. AFOG effectiveness over 12 detection transformers, measured by mAP on perturbed images. (*) indicates DINO framework used with the corresponding backbone for object detection.

| Model | Params (M) | Benign | AFOG | AFOG-V | AFOG-F |
|---|---|---|---|---|---|
| DETR-R50 [4] | 39.8 | 42.1 | 4.1 | 4.5 | 9.8 |
| DETR-R101 [4] | 76.0 | 43.5 | 5.2 | 5.1 | 11.3 |
| Deform.-DETR [39] | 40.0 | 44.5 | 4.8 | 1.5 | 7.1 |
| R50* [12] | 47.6 | 49.2 | 5.3 | 1.5 | 6.3 |
| AlignDETR [2] | 47.6 | 51.4 | 18.1 | 1.6 | 1.4 |
| ViTDet* [15] | 108.1 | 54.9 | 3.8 | 0.9 | 2.8 |
| ConvNeXt* [22] | 219.0 | 55.4 | 3.9 | 1.9 | 3.1 |
| Swin-L* [21] | 217.2 | 56.8 | 7.3 | 2.4 | 8.6 |
| InternImage* [32] | 241.0 | 56.9 | 7.3 | 2.8 | 5.1 |
| FocalNet* [37] | 228.9 | 58.5 | 7.3 | 2.5 | 5.1 |
| EVA* [11] | 1037.2 | 62.1 | 12.2 | 4.1 | 8.7 |
| DETA [26] | 218.8 | 62.9 | 25.6 | 3.7 | 4.3 |

12개의 Transformer Detector를 이용한 실험 결과

대부분 Bengin -> 공격 후 mAP가 줄어든 것을 확인

# Experiments

Table 2. Comparison benchmark of AFOG against other state-of-the-art object detection attacks on DETR and Swin. Results are theirs. (*) indicates results from [33]. (†) indicates results from [25]. (-) indicates no result.

| Attack | Type | Pert. Budget | Iters. | Adversarial mAP DETR-R50 | Adversarial mAP Swin |
|---|---|---|---|---|---|
| GARSDC [18] | Surrogate | 0.05 | 3000+ | 6.0 | - |
| GALD [14] | Surrogate | 0.063 | 10 | 20.6 | - |
| RAD* [6] | Surrogate | 0.063 | 10 | 27.2 | 47.2 |
| GHFD* [33] | Surrogate | 0.063 | 50 | 12.7 | 42.3 |
| UEA* [34] | Surrogate | 0.063 | 50 | 28.5 | 50.7 |
| DAG* [36] | Surrogate | 0.063 | 50 | 28.6 | 50.7 |
| RAP* [16] | Surrogate | 0.063 | 50 | 24.7 | 49.5 |
| EBAD† [3] | Victim | 0.039 | 10 | 34.9 | - |
| AttentionFool [23] | Victim | - | 10-150 | 21.0 | - |
| OATB [13] | Victim | 0.078 | 20 | 26.6 | - |
| DBA [17] | Victim | - | 50 | - | 56.7 |
| **AFOG** | **Victim** | **0.031** | **10** | **4.1** | **7.3** |

Black Box At-
tack

White Box At-
tack

AFOG At-
tack

기존 Adv Attack 실험 결과 비교

대부분 Bengin -> 공격 후 mAP가 줄어든 것을 확인

# Experiments

Table 3. Timing and Imperceptibility Results. $L_2$ represents the average $L_2$ norm difference between perturbed and clean images, $L_0$ is the average proportion of perturbed pixels, SSIM is the structural similarity index measure, $\mu_\triangle$ is the average perturbation magnitude, and t is average total attack time for all ten iterations in seconds.

| | AFOG | | | | | AFOG-V | | | | | AFOG-F | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model | $L_2$ | $L_0$ | SSIM | $\mu_\triangle$ | time | $L_2$ | $L_0$ | SSIM | $\mu_\triangle$ | time | $L_2$ | $L_0$ | SSIM | $\mu_\triangle$ | time |
| DETR-R50 [4] | 0.0322 | 0.9707 | 0.8715 | 0.0173 | 1.45 | 0.0323 | 0.9707 | 0.8716 | 0.0172 | 0.99 | 0.0323 | 0.9710 | 0.8717 | 0.0172 | 1.21 |
| DETR-R101 [4] | 0.0323 | 0.9707 | 0.8721 | 0.0173 | 1.47 | 0.0323 | 0.9706 | 0.8724 | 0.013 | 1.16 | 0.0323 | 0.9708 | 0.8724 | 0.0172 | 1.70 |
| Deform.-DETR [39] | 0.0323 | 0.9719 | 0.8711 | 0.0174 | 1.63 | 0.0323 | 0.9713 | 0.8716 | 0.0173 | 1.63 | 0.0012 | 0.9714 | 0.8717 | 0.0173 | 1.87 |
| R50 [12] | 0.0317 | 0.9658 | 0.8343 | 0.0171 | 2.70 | 0.0317 | 0.9650 | 0.8348 | 0.0170 | 2.34 | 0.0317 | 0.9654 | 0.8346 | 0.0170 | 3.16 |
| AlignDETR [2] | 0.0319 | 0.9657 | 0.8347 | 0.0170 | 2.41 | 0.0319 | 0.9646 | 0.8349 | 0.0170 | 2.33 | 0.0319 | 0.9647 | 0.8349 | 0.0170 | 3.29 |
| ViTDet [15] | 0.0318 | 0.9671 | 0.8353 | 0.0171 | 6.88 | 0.0318 | 0.9657 | 0.8361 | 0.0170 | 6.67 | 0.0318 | 0.9664 | 0.8355 | 0.0171 | 7.33 |
| ConvNext [22] | 0.0318 | 0.9666 | 0.8342 | 0.0171 | 5.38 | 0.0318 | 0.9654 | 0.8349 | 0.0170 | 5.26 | 0.0318 | 0.9663 | 0.8347 | 0.0170 | 5.86 |
| Swin-L [21] | 0.0327 | 0.9724 | 0.8673 | 0.0175 | 7.13 | 0.0327 | 0.9716 | 0.8680 | 0.0173 | 7.28 | 0.0327 | 0.9722 | 0.8678 | 0.0174 | 8.99 |
| InternImage [32] | 0.0318 | 0.9665 | 0.8360 | 0.0170 | 6.35 | 0.0318 | 0.9653 | 0.8367 | 0.0170 | 6.23 | 0.0318 | 0.9660 | 0.8364 | 0.0171 | 6.70 |
| FocalNet [37] | 0.0320 | 0.9665 | 0.8365 | 0.0172 | 8.96 | 0.0320 | 0.9657 | 0.8378 | 0.0171 | 8.84 | 0.0320 | 0.9659 | 0.8374 | 0.0171 | 9.74 |
| EVA [11] | 0.0370 | 0.9666 | 0.8240 | 0.0172 | 54.34 | 0.0370 | 0.9665 | 0.8246 | 0.0171 | 54.53 | 0.0370 | 0.9665 | 0.8242 | 0.0171 | 51.18 |
| DETA [26] | 0.0481 | 0.9662 | 0.8096 | 0.0170 | 13.20 | 0.0481 | 0.9654 | 0.8099 | 0.0170 | 13.10 | 0.0481 | 0.9654 | 0.8099 | 0.0170 | 13.13 |

SSIM : 이미지 구조적 유사도 (1에 가까울 수록 유사함)

Time : Adv Attack 10회 반복하는데 걸린 시간

# Experiments

Table 4. Comparing AFOG with four state-of-the-art attacks on representative CNN-based object detectors. mAPs of existing attacks were taken from respective papers [8]. (-) indicates N/A.

| Model | Attack | mAP | t | Distortion Cost $L_\infty$ | Distortion Cost $L_2$ | Distortion Cost $L_0$ | Distortion Cost SSIM |
|---|---|---|---|---|---|---|---|
| YOLOv3 | Benign | 83.43 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 |
| | TOG [8] | **0.56** | **0.98** | 0.031 | 0.083 | 0.984 | **0.875** |
| | AFOG | 2.28 | 1.31 | 0.031 | **0.013** | **0.855** | 0.801 |
| SSD-300 | Benign | 76.11 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 |
| | UEA [34] | 20.0 | - | - | - | - | - |
| | DAG [36] | 64.0 | - | - | - | - | - |
| | TOG [8] | 0.86 | **0.39** | 0.031 | 0.120 | 0.975 | **0.879** |
| | AFOG | **0.50** | 0.49 | 0.031 | **0.022** | **0.858** | 0.793 |
| FRCNN | Benign | 67.37 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 |
| | UEA [34] | 5.0 | 0.17 | 0.343 | 0.191 | 0.959 | 0.652 |
| | RAP [16] | 4.78 | 4.04 | 0.082 | 0.010 | 0.531 | 0.994 |
| | DAG [36] | 3.56 | 7.99 | **0.024** | **0.002** | **0.493** | **0.999** |
| | TOG [8] | 2.64 | **1.68** | 0.031 | 0.058 | 0.976 | 0.862 |
| | AFOG | **2.38** | 2.11 | 0.031 | 0.019 | 0.854 | 0.788 |

CNN Based object Detector Experiments

CNN Based 에도 효과적임을 보임(전이성이 좋음)

# Conclusion

- We proposed **AFOG**, a novel white-box attack for analyzing object detectors.
- It provides a **unified, architecture-agnostic framework** for both Transformers and CNNs.
- Its key, **'Learnable Attention'**, effectively finds and focuses on the most vulnerable areas.
- It achieves high **efficiency and stealth** by using minimal, imperceptible perturbations.
- Experiments show AFOG **outperforms SOTA methods** in effectiveness, stealth, and speed.