Distortion-Aware Adversarial Attacks on Bounding Boxes of Object Detectors

Pham Phuc[1], Son Vuong[2,3], Khang Nguyen[4] and Tuan Dang[4]
[1]*Ho Chi Minh City University of Technology, Vietnam*
[2]*VinBigData, Vietnam*
[3]*VNU University of Engineering and Technology, Vietnam*
[4]*University of Texas at Arlington, U.S.A.*

Keywords: Adversarial Attacks, Object Detection, Model Vulnerability.

Abstract: Deep learning-based object detection has become ubiquitous in the last decade due to its high accuracy in many real-world applications. With this growing trend, these models are interested in being attacked by adversaries, with most of the results being on classifiers, which do not match the context of practical object detection. In this work, we propose a novel method to fool object detectors, expose the vulnerability of state-of-the-art detectors, and promote later works to build more robust detectors to adversarial examples. Our method aims to generate adversarial images by perturbing object confidence scores during training, which is crucial in predicting confidence for each class in the testing phase. Herein, we provide a more intuitive technique to embed additive noises based on detected objects' masks and the training loss with distortion control over the original image by leveraging the gradient of iterative images. To verify the proposed method, we perform adversarial attacks against different object detectors, including the most recent state-of-the-art models like YOLOv8, Faster R-CNN, RetinaNet, and Swin Transformer. We also evaluate our technique on MS COCO 2017 and PASCAL VOC 2012 datasets and analyze the trade-off between success attack rate and image distortion. Our experiments show that the achievable success attack rate is up to 100% and up to 98% when performing white-box and black-box attacks, respectively. The source code and relevant documentation for this work are available at the following link https://github.com/anonymous20210106/attack_detector.git.

1 INTRODUCTION

Neural network-based detectors play significant roles in many crucial downstream tasks, such as 3D depth estimations (Dang et al., 2023), 3D point cloud registration (Nguyen et al., 2024a), semantic scene understanding (Nguyen et al., 2024b), and visual SLAM (Dang et al., 2024). However, neural networks are proven to be vulnerable to adversarial attacks, especially for vision-based tasks. Starting from image classification, prior works (Goodfellow et al., 2015; Madry et al., 2018; Moosavi-Dezfooli et al., 2016) try to attack classification models systematically. Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2015) and Projected Gradient Descent (PGD) (Madry et al., 2018) leverage gradients of the loss function to add a minimal perturbation and find the direction to move from the current class to the targeted class. In this realm, DeepFool (Moosavi-Dezfooli et al., 2016) formed this as an optimization problem to find both minimal distances and optimal direction by approximating a non-linear classification using the first order of Taylor expansion and Lagrange multiplier. Besides gradient-based approaches, (Alaifari et al., 2018) generated adversarial images by optimizing deformable perturbation using vector fields of the original one. Although adversarial attacks gain more attention and effort from researchers, crafting theories and practical implementation for this problem on object detectors are not well-explored compared to itself on classification tasks.

Motivated by adversarial attacks for classification, recent works (Song et al., 2018; Lu et al., 2017; Im Choi and Tian, 2022; Xie et al., 2017) attempt to perturb image detectors. Patch-based approach (Lu et al., 2017; Song et al., 2018; Liu et al., 2019; Du et al., 2022) adds random patches or human design patches into original images; these methods are reported to be effective in fooling the detectors, but the patches are apparently visible to the human eyes.

Phuc, P., Vuong, S., Nguyen, K. and Dang, T.
Distortion-Aware Adversarial Attacks on Bounding Boxes of Object Detectors.
DOI: 10.5220/0013101900003912

In *Proceedings of the 20th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2025) - Volume 2: VISAPP*, pages 27-38
ISBN: 978-989-758-728-3; ISSN: 2184-4321

Figure 1: **The adversarial attack on bounding boxes of object detectors with distortion awareness** can perturb a sequence of images taken from a surveillance camera with a controllable added amount of distortion to obtain a certain success attack rate (Sec. 3 and Sec. 4), making the object detector disabled. The demonstration video of the illustrated sequence with more examples is available at YouTube.

Dense Adversary Generation (DAG) (Xie et al., 2017) considers fooling detectors as a fooling classifier for proposed bounding boxes: perturbing labels in each proposed bounding box to make the detector predict a different label other than the true one. Meanwhile, another method (Im Choi and Tian, 2022) focuses on attacking location, objectness, or class confidence of YOLOv4 (Bochkovskiy et al., 2020) by noising the vehicle-related image using FGSM and PGD methods. However, this technique lacks the knowledge of individual bounding boxes, resulting in inaccurate added noises in multi-object images. Furthermore, the quality of adversarial images is not well-studied and is often disregarded in previous works due to their prioritization of attacking methods' effectiveness.

Indeed, perturbing object detectors is far more challenging since the network abstracts location regression and object confidence, and loss functions are often multi-task learning. Self-exploring images and finding the best perturbation like (Moosavi-Dezfooli et al., 2016; Alaifari et al., 2018) for detectors become exhausting because of multi-task learning. As learning to detect objects in images heavily depends on the objective functions or loss functions, the objective of training detectors is to minimize and converge these losses. Thus, one way to attack detectors is to increase losses for training samples to a certain level so that detectors misdetect or no longer recognize any objects. Through this observation, our approach is to find the optimal direction and distortion amount added to the targeted pixels with respect to these losses. Fig.1 demonstrates the practical application of our distortion-aware adversarial attack technique in real-world surveillance scenarios. The method introduces adversarial perturbations that cannot be recognized by humans but effectively disable object detection systems. It maintains a balance between preserving image quality and achieving high attack success rates, making it flexible across various practical situations. The unnoticeable nature of these distortions is crucial for adversarial use cases, as they remain visually undetectable while exploiting weaknesses in modern object detection models. This combination of stealth and effectiveness highlights the robustness of our approach.

To implement our method, we leverage the gradient from the loss function, like FGSM. While FGSM adds the exact amount of noise to every pixel except ones that do not change their direction, our approach uses magnitude from the gradient to generate optimal perturbations to all targeted pixels. As detectors propose bounding boxes and predict if objects are present in such regions before predicting which classes they belong to, object confidence plays an essential role in the detection task. We, therefore, inclusively use these losses and further sampling with a recursive gradient to take advantage of valuable information from all losses. We also find optimal perturbation amount iteratively as iterative methods produce better results than the fast methods.

In this work, our contributions are summarized as follows: (1) formalize a distortion-aware adversarial attack technique on object detectors, (2) propose a novel approach to attack state-of-the-art detectors with different network architectures and detection algorithms (Ren et al., 2015; Lin et al., 2017; Liu et al., 2021; Jocher et al., 2023), and (3) analyze and experiment our proposed technique on MS COCO 2017 (Lin et al., 2014) and PASCAL VOC 2012 (Everingham et al., 2015) datasets with cross-model transferability and cross-domain datasets validation. Our key properties compared to previous methods (Xie et al., 2017; Wei et al., 2019) are also shown in Tab.1.

Table 1: Comparisons between our method and previous methods, including DAG (Xie et al., 2017) and Unified and Efficient Adversary (UEA) (Wei et al., 2019), in terms of key properties.

	DAG	UEA	Ours
iterative added noises	✓	✗	✓
mostly imperceptible to human eyes	✓	✓	✓
distortion awareness	✗	✗	✓
stable transferability to backbones	✗	✗	✓
consistent with detection algorithms	✗	✗	✓

2 RELATED WORK

Adversarial Attacks on Object Detectors. Previous works in adversarial attacks on object detection can be categorized into optimization problems and Generative Adversarial Networks (GAN). The optimization problem is finding the adversarial images that satisfy the objective functions, while GAN generates adversarial images by training a generator that focuses on a classification or regression of the target network (Wei et al., 2019). Other methods use patches to fool the detectors (Song et al., 2018; Du et al., 2022), but noises are visible from a human perspective. We consider the adversarial attack as an optimization problem. Our method is conceptually similar to DAG (Xie et al., 2017), but we more focus on finding the optimal direction and amplitude for each pixel to perturb given bounding boxes rather than drifting from one true class to another while proposing bounding boxes, which is impractical when class labels are unknown, especially in black-box attacks. Furthermore, we demonstrate the effectiveness of our methods on both one-stage and two-stage detectors.

Iterative Generative of Adversarial Images. Inspired by the earliest study on classification problems (Goodfellow et al., 2015), the work (Kurakin et al., 2018) shows the effectiveness of iterative methods over one-shot methods by using the least-likely class method with FGSM to generate adversarial images for classification tasks. Another work (Alaifari et al., 2018) iteratively adds small deformation constructed by vector fields into images while DAG (Xie et al., 2017) performs iterative gradient back-propagation on adversarial labels for each target. Our method also uses iterative methods; however, differs from the mentioned methods: we calculate the gradient over the iteratively permuted images and optimize this gradient under image distortion control. Moreover, we also focus on attacking general image detectors at different network architectures and detection methods, while (Goodfellow et al., 2015; Kurakin et al., 2018; Alaifari et al., 2018) focus on attacking classifiers.

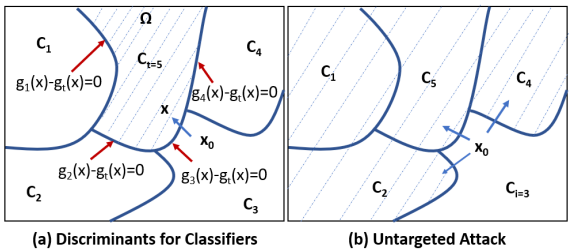

Figure 2: Illustration of adversarial attack with decision boundaries formed by k discriminant functions: attackers are looking for alternative x that is similar to x_0 such that $g_i(x) < g_t(x_0)$ for $i = 1,2,..,k$ and $t \neq i$ so that the model f classify x as t. Untargeted attack is seeking x such that the model, f, classifies x as all C_j where $i \neq j$. In this example, $t = 5$ and $i = 3$

Image Distortion Measurement. Prior works (Kurakin et al., 2018; Carlini and Wagner, 2017; Chen et al., 2018) used l^∞ to measure the similarity between images original and adversarial images. l^∞ effectively associates the corresponding features between pairs of images under changes, such as shifting or rotations (Wang et al., 2004; Lindeberg, 2012; Rublee et al., 2011). Regardless, as l^∞ focuses on a per-pixel level, it lacks the illustration of how changes in a pixel might affect its neighboring pixels or might impact the overall pattern of the distorted image (Puccetti et al., 2023). Other methods, such as mean square error (MSE), peak signal-to-noise ratio (PSNR), and contrast-to-noise ratio (CNR) are less sensitive to the human visual system (Lu, 2019). Therefore, we select Normalize Cross Correlation, which is robust to various image scales and less computational than Structural Similarity (SSIM) (Wang et al., 2004) while maintaining the distortion imperceptibility to human perception.

3 FORMULATION

This section formalizes the attack strategy through key equations, including perturbation minimization (Eq.1), discriminant functions (Eq.3), and optimization objectives (Eq.6).

3.1 Adversarial Attacks on Object Detectors

Defintion. Let I be an RGB image of size of $m \times n \times 3$ with objects, $o_1, o_2, ..., o_k$, belonging to classes, $c_1, c_2, ..., c_k$. Similarly, the perturbed image is denoted as $I^{'}$ but with the corresponding classes are now $c_1^{'}, c_2^{'}, ..., c_k^{'}$, where $\{c_1, c_2, ..., c_k\} \neq \{c_1^{'}, c_2^{'}, ..., c_k^{'}\}$.

Therefore, our objective is to identify an algorithm such that the difference between I and $I^{'}$ is minimized, so that $I^{'}$ can still perturb the detector, f, to misdetect objects but is mostly imperceptible to human eyes. The procedure, with ε as the distortion (perturbation) amount, is written as:

$$\underset{\varepsilon}{\text{minimize}} \; ||I - I^{'}||, \text{ with } \varepsilon = I^{'} - I \tag{1}$$

Discriminants for Classifiers. The decision boundaries between a k-class-classifier are formed by k discriminant functions, $g_i(\cdot)$, with $i = 1, 2, ..., k$, as illustrated in Fig.2a. Also, for untargeted attacks, misdetecting a particular object in $I^{'}$ requires moving $f(o_i)$ into a class other than its true class, $c_i^{'}$, as shown in Fig.2b. Thus, the domain, Ω_i, that $f(o_i)$ results in $c_i^{'}$ is defined as follows:

$$\Omega_i \coloneqq \left\{ o_i \mid g_i(o_i) - \min_{j \neq i} \{g_j(o_i)\} \leq 0 \right\} \tag{2}$$

Discriminants for Object Detectors. Moreover, in the scope of object detection, accurate detections mainly rely on *the class confidence scores of objects in bounding boxes* after non-max suppression. Therefore, the class confidence score should be inferred to be less than the confidence threshold for the detector to misdetect classes of objects in the bounding boxes. Reforming Eq.2, we obtain:

$$\Omega_i \coloneqq \left\{ o_i \mid p(c_i) - \min_{j \neq i} \{p(c_j)\} \leq \mathcal{T} \right\} \tag{3}$$

for $b_i \in \{b_1, ..., b_k\}$ and $\{b_1, ..., b_k\} \sim \{o_1, ..., o_k\}$

with $\sim$ represents the element-wise corresponding notation, $\{b_1, b_2, ..., b_k\}$ indicate the detected boxes in I, $\mathcal{T}$ is the pre-defined confidence threshold, and $p(\cdot)$ represents the class probability function.

3.2 Perturbing to Change Class Confidence Scores

Class Confidence Score. To change the class confidence score of an object in a bounding box, we perturb its likelihood, $Pr(c_i \mid o_i)$, to bring $p(c_i)$ to be lower than the class probability, $p(c_j)$, and the likelihood of another class, $Pr(c_j \mid o_i)$, as formalized as follows:

$$\begin{aligned} p(c_i) &= Pr(c_i \mid o_i) \cdot Pr(o_i) \\ &< Pr(c_j \mid o_i) \cdot Pr(o_i) = p(c_j) \end{aligned} \tag{4}$$

In short, to do this, the adversarial distortions should be added in each proposed bounding box. Therefore, based on Eq.4 and $Pr(o_i) \geq 0$ meaning that there is a chance that the object is presented in the bounding box, Eq.3 therefore can be rewritten into:

$$\Omega_i \coloneqq \left\{ o_i \mid Pr(c_i \mid o_i) - \min_{j \neq i} \{Pr(c_j \mid o_i)\} \leq \mathcal{T} \right\} \tag{5}$$

Objective Function for Object Detection. Combining Eq.1 and Eq.5, the generalized optimization generating an adversarial image that perturbs f to misdetect o_i in b_i within I is defined as follows:

$$\underset{\varepsilon}{\text{minimize}} \; ||I - I^{'}|| \; \underline{\text{such that}} \; \Omega_i \leq \mathcal{T} \tag{6}$$

3.3 Perturbing Through Detector Loss

Detector Loss: Most commonly-used object detectors return predicted classes with their corresponding bounding box coordinates and confidence scores. In which, the loss function, $\mathcal{L}$, is the sum of classification loss, $\mathcal{L}_{cls}$, localization loss, $\mathcal{L}_{loc}$, and confidence loss, $\mathcal{L}_{obj}$, as below:

$$\mathcal{L} = \mathcal{L}_{loc} + \mathcal{L}_{obj} + \mathcal{L}_{cls} \tag{7}$$

Perturbing through Detector Loss. Based on Eq.7, to *desired target pixels* to perturb in an image, we add the amount of distortion as follows:

$$\frac{\partial \mathcal{L}}{\partial I} \cdot \mathbf{M}[f(I)] \tag{8}$$

where $\mathbf{M}$ represents all masks predicted by f on I, which is the sum of bounding boxes on an m-by-n zeroes array, and ∂ indicates the partial derivative notation.

Therefore, to perturb the classes' probabilities of an object in a bounding box, we can instead modulate it through the definition in Eq.8, which effectively fools the object detectors during the inference stage. The involvement of Eq.8 is shown in Eq.9 (Sec.4.1).

4 METHOD

In this section, we propose the white-box attack algorithm (Sec.4.3) to find the most appropriate distortion amount, ε, via generating adversarial images, $I^{'}$, iteratively (Sec.4.1) with distortion awareness (Sec.4.2).

4.1 Iterative Adversarial Images

With the assumption that the object detector's network architecture is known, our proposed method leverages the gradient of how pixels of predicted objects change when I passes through the network. In specific, we find the gradient ascent of targeted pixels to convert the original image, I, to an adversarial image, $I^{'}$.

Generating Iterative Adversarial Images. However, the gradient derived from the total loss (Eq.7) also gives the gradient of non-interested regions; meanwhile, we need to navigate the adversarial image to follow the gradient on specific bounding boxes. Using Eq.8, we search for the adversarial image with respect to the gradient ascent of targeted pixels as follows:

$$I_i^{'} = I_{i-1}^{'} + \varepsilon = I_{i-1}^{'} + \lambda \cdot \frac{\partial \mathcal{L}}{\partial I_{i-1}^{'}} \cdot \mathbf{M}\left[f(I_{i-1}^{'})\right] \quad (9)$$

$$\text{with } \mathbf{M}\left[f(I_{i-1}^{'})\right] = \mathbf{0}_{m\times n} + \sum_{i=1}^{k} b_i \text{ and } I_0^{'} = I$$

where the subscripts, i and $i-1$, denote current and previous iterations, respectively, $+$ sign denotes the gradient direction (ascending), and λ is the gradient ascent's step size.

Distortion as Control Parameter. Iterating Eq.9 over a considerable iterations, the generated adversarial image, $I^{'}$, might get over-noised, which dissatisfies Eq.1 and eventually Eq.6 regarding minimizing ε. Therefore, we introduce two strategies to control the distorted images:

$$I^{'} = \begin{cases} I_i^{'}, & \text{if } \mathcal{D}(I, I_i^{'}) \geq \mathcal{S} \text{ or } f(I_i^{'}) \geq \mathcal{R} \\ I_{i+1}^{'}, & \text{otherwise (using Eq.9)} \end{cases} \quad (10)$$

where $\mathcal{D}(I, I_i^{'})$ computes the distortion amount, ε, between I and $I_i^{'}$ as subsequently defined in Eq.12 (Sec.4.2), and $\mathcal{S}$ and $\mathcal{R}$ are the target distortion amount and the desired success attack rate, respectively, which are variants of $\mathcal{T}$.

Differences of Proposed Strategies. Both conditional statements of Eq.10 eventually help Eq.9 to find the smallest iteration without brute-forcing over a larger iteration. Yet, the main difference between these equations is that $\mathcal{D}(\cdot,\cdot) \geq \mathcal{S}$ focuses on adding a desired distortion in the original image. Meanwhile, $f(\cdot) \geq \mathcal{R}$ concentrates on the desired success attack rate. Eq.9 is the extended version applied for detectors derived from (Kurakin et al., 2018).

4.2 Normalized Cross Correlation

As Normalized Cross Correlation (NCC) depicts abrupt changes of targeted pixels to the average value of all image pixels while computing the similarity between two input images, we use NCC for our work, as shown in Eq.11.

$$\text{NCC}(I, I^{'}) = \frac{\sum_{i=1}^{n}\left(I_{(i)} - \overline{I}\right)\left(I_{(i)}^{'} - \overline{I^{'}}\right)}{\sqrt{\sum_{i=1}^{n}\left(I_{(i)} - \overline{I}\right)^2}\sqrt{\sum_{i=1}^{n}\left(I_{(i)}^{'} - \overline{I^{'}}\right)^2}} \quad (11)$$

Algorithm 1: Adversarial Images Iterative Generation.

Input : $I :=$ raw image, $\lambda :=$ step size
$f :=$ detection model, $N :=$ max iteration
$\mathcal{T} = \{\mathcal{S} \mid \mathcal{R}\} :=$ control param
Output: $I^{'} :=$ adversarial image

function generator$(I, \lambda, f, N, \{\mathcal{S} \mid \mathcal{R}\})$
  $i = 0$, $I_i^{'} = I$
  $\{b_1, b_2, ..., b_k\} = \mathbf{B}[f(I)]$
  while $i < N$ ***and*** $\{b_1, b_2, ..., b_k\} \neq \emptyset$ **do**
    $\mathbf{M} = \mathbf{0}_{m\times n}$
    $\{b_1^{'}, b_2^{'}, ..., b_k^{'}\} = \mathbf{B}\left[f(I_i^{'})\right]$
    for $b_j^{'} \in \{b_1^{'}, b_2^{'}, ..., b_k^{'}\}$ **do**
      if $\mathcal{D}(I, I_i^{'}) \geq \mathcal{S}$ ***or*** $f(I_i^{'}) \geq \mathcal{R}$ **then**
        break
      $\mathbf{M} \leftarrow \mathbf{M} + b_j^{'}$
    $I_{i+1} = I_i + \lambda \cdot \frac{\partial \mathcal{L}}{\partial I_i^{'}} \cdot \mathbf{M}\left[f(I_i^{'})\right]$ (Eq.9)
    $i \leftarrow i + 1$
  $I^{'} = I_i^{'}$
  return $I^{'}$

with n is the number of pixels in I and $I^{'}$, $I_{(i)}$ indicates the i^{th} pixel of I, and $\overline{I}$ represents the mean value of I.

Since $\text{NCC}(I, I^{'}) \in [0,1]$ measures the similarity score between I and $I^{'}$, we define the distortion metric (dissimilarity), $\mathcal{D}$, as the complement of NCC in 1, as follows:

$$\mathcal{D}(I, I^{'}) = 1 - \text{NCC}(I, I^{'}) \quad (12)$$

4.3 Algorithm

As illustrated in Alg.1, the algorithm first takes the bounding boxes that predicted objects provided by f on a raw image I. Hence, the adversarial image generation takes place iteratively until the predefined maximum iteration, M, is reached or no bounding boxes on $I_i^{'}$ are detected by f. As the bounding boxes are re-predicted in each iteration, ε is added based on the change of $\mathcal{L}$ with respect to the pixel's gradient ascent of $I_i^{'}$. Using on Eq.9, ε is only added on the aggregated masks, $\mathbf{M}\left[f(I_i^{'})\right]$, of the bounding boxes. To better control either ε to be added or the success attack rate, $\mathcal{R}$, we also check if $\mathcal{D}(I, I_i^{'})$ or $f(I_i^{'})$ exceeds the predefined threshold (Eq.10) to maintain the adversarial image to be adequately controlled; otherwise, ε are kept adding in the next iteration. Note that the conditional statements in Alg.1 can be used independently, which either controls $\mathcal{R}$ or $\mathcal{S}$. The analyses on control of $\mathcal{R}$ and $\mathcal{S}$ with respect to $I_i^{'}$ are further provided in Sec.5.

5 ANALYSES

To verify our proposed method's attacking feasibility, we analyze it with a subset of images on the *most recent state-of-the-art* detection models (YOLOv8 – with various sizes).

5.1 Convergence of Losses

The total loss consistently converges as adversarial images are iteratively generated, as shown in Fig.3. To validate this behavior, we conducted extensive testing on numerous images from the MS COCO 2017 dataset, confirming that the convergence trend is consistent across the entire dataset. For visualization purposes, we randomly selected three representative images to illustrate this trend. Through our experiments, we found that 120 iterations strike an optimal balance between computational efficiency and attack performance, allowing sufficient time for the total loss to converge. This iteration count ensures that the results are representative and practical for real-world applications.

This also shows that Alg.1 can find adversarial images that can fool the object detectors. Also, the distortions of the adversarial images become larger as the iterations increase. Therefore, if we pick a recursively adversarial image before the convergence, we get a less-distorted image but eventually sacrifice the effectiveness of our attack.

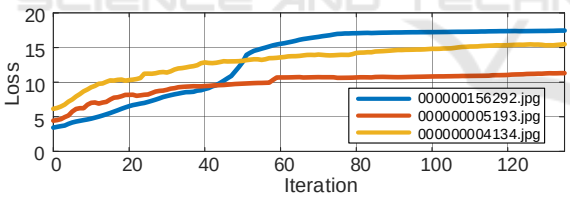

Figure 3: The convergence of loss over 120 iterations on a subset of images from the MS COCO 2017 dataset.

5.2 Image Distortion for Difference Models with Confidence Thresholds and Success Rate

Success Rates. Fig.4 shows that YOLOv8n is the most vulnerable model with the least distorted image. In contrast, YOLOv8x is the hardest to attack, and its adversarial images are the most distorted compared to other models. Indeed, we can achieve a success attack rate of more than 80% if image distortion is set by 10%. However, if the distortion rate increases from 10%, the attacking rate increases slowly. Overall, we can obtain a decent attacking rate by distorting only parts of images.

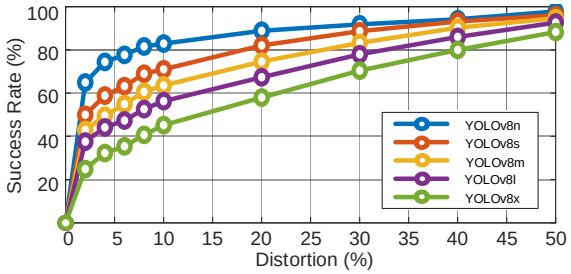

Figure 4: Relationship between attacking rate and target distortion on detection models set with confidence thresholds of 0.75.

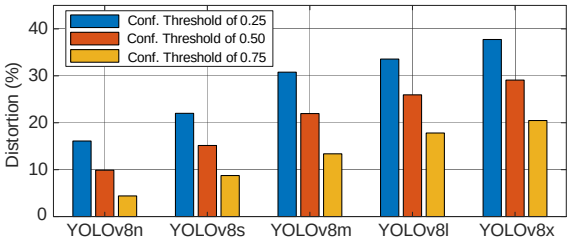

Figure 5: Relationship between confidence score and distortion at a success attack rate of 97% on various-sized YOLOv8 models.

Confidence Scores. We also evaluate Alg.1 to see how average image distortion changes for each model when obtaining a desired attacking rate. Fig.5 depicts that attacking models with lower confidence scores causes the original images to be distorted more than the same model set with higher confidence scores.

5.3 Distortion Amount and Number of Iterations to Fool Different-Sized Models

Distortion Amount. The bottom row of Fig.9 shows the added distortion amounts (*top row*) to generate the adversarial images (*middle row*) among various-sized models. We notice that, for larger-sized models, our method tends to add more noise to prevent these models from extracting the objects' features and thereafter recognizing them, and vice versa. In this case, the features of the bear are perturbed. Another noticeable point is that the added distortion amount becomes more visible to human eyes when fooling the large-sized models, as depicted in the adversarial images and the heatmaps in the last two columns of Fig.9.

Number of Iterations. As proven that our method needs more iterations to generate noise to fool large models, we also provide the number of iterations to generate such perturbations, as shown in Fig.6, which shows the approximately-proportional trend between the number of iterations to the sizes of models.

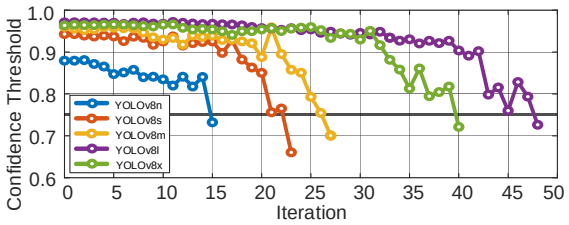

Figure 6: The needed (minimum required) iterations for Alg.1 to fool various-sized models with confidence thresholds of 0.75.

5.4 Success Rate of Adversarial Images to Models with Different Confidence Thresholds

Since different detectors are often set with different confidence thresholds, we analyze how many iterations our method takes to obtain the success attack rate of 100%, where all objects presented in the image are misdetected. Fig.7 (*left*) shows the increasing effectiveness of noises added to the raw image over 90 iterations. In fact, with a confidence threshold of 0.50, the detector is unable to detect objects in the image; meanwhile, the detector with a confidence threshold of 0.25 can still detect objects, but the detections become inaccurate. starting from the 75^{th} iteration. However, this particular image only illustrates results that the bounding boxes are not overlapped with each other.

Fig.7 (*right*) also shows the results where objects are overlapped with each other: the orange's bounding box is in the person's bounding box. Our method also obtains the success attack rate of 100% to the model with the confidence threshold of 0.50. Nevertheless, this process takes about 580 iterations to completely fool the detector.

5.5 Attention of Detection Models

To further explain our method, we analyze how the model's attention altered using Grad-CAM (Selvaraju et al., 2017), as illustrated in Fig.8. Before being attacked (Fig.8a), the model is able to detect objects with high confidence scores, and its attention map (Fig.8b) accurately focuses on the areas presumed to contain objects. However, while performing Grad-CAM on perturbed images, the model fails to detect objects surpassing the confidence threshold (Fig.8d). Moreover, the model identifies the segmented regions, as visualized on attention maps, belonging to different classes.

Also, as mentioned in Sec.4.1, our method strives to decrease the confidence scores of objects in each bounding box by determining the optimal noises, resulting in changes in the model's attention and, thereafter, its detection. Indeed, the attention map focuses on the same bounding boxes, and their intensities change since the confidence scores are reduced significantly, leading to misdetection.

Table 2: Comparisons of success attack rates between DAG (Xie et al., 2017) and our method on detection models with **ResNet-50 backbone**.

	ResNet-50 Backbone			
	Faster R-CNN	RetinaNet	Swin-T	R-FCN -RN50
Baseline	27.20	22.90	32.47	76.40
DAG (Xie et al., 2017)	-	-	-	63.93
Ours	5.32	3.58	8.57	-
Succ. Rate	80.44%	84.37%	73.61%	16.32%

Analysis Conclusions. Our analyses allude that larger models might easily overcome adversarial attacks; however, this also raises the concern of computing power while training these large-sized models with adversarial examples and deploying them for real-world applications.

6 EXPERIMENTS

We evaluate our proposed method on MS COCO 2017 (Lin et al., 2014) and PASCAL VOC 2012 (Everingham et al., 2015) datasets with other detection algorithms of different backbones, including validating with cross-model and cross-domain datasets and verifying their transferability to different backbones and consistency with different detection algorithms. In specific, the experiments are conducted as follows: (1) generating adversarial images against one detector, then (2) perturbing other detectors using those images without prior knowledge about the models.

6.1 Cross-Model Validation

We use pre-trained models (YOLOv8, Faster-RCNN, RetinaNet, Swin Transformer) trained on MS COCO 2017 and generate adversarial examples for each model on the validation set of MS COCO 2017. The adversarial examples generated by one model are evaluated by others, including itself. Tab.4 shows that models are fooled by adversarial images generated by themselves, in which these images actually include knowledge of that model: the most optimal (best) perturbation to make that specific model misdetect.

The results also show that the larger-sized models generate adversarial examples that are more effective against smaller ones. Notably, also from Tab.4, our method best performs when testing its adversarial examples (against YOLOv8x) on other models since it produces more generalized noises affecting

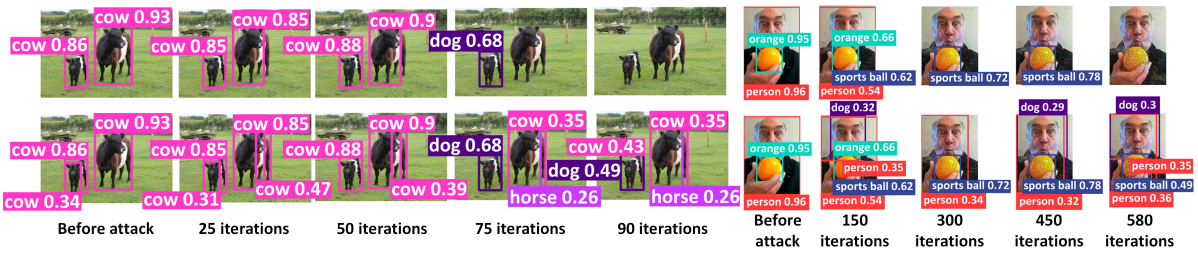

Figure 7: Adversarial images generated by Alg.1 at different iterations and **how they affect the detector's performance** at confidence thresholds of **0.50 (top)** and **0.25 (bottom)**, respectively. The case of non-overlapping bounding boxes (*left*) effectively causes the detector to recognize the wrong objects before misdetecting objects at the 90^{th} iteration at a confidence threshold of 0.50. Compared to the case where overlapped bounding boxes exist (*right*), Alg.1 takes more iterations (580 iterations) to fool the detector with the same configuration.

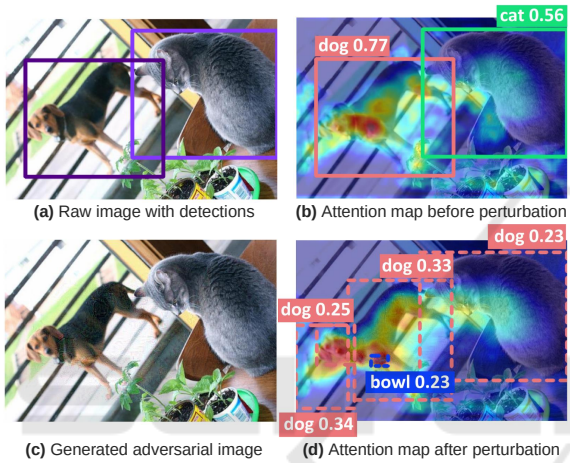

Figure 8: Visualization on attention maps before and after perturbation. The confidence scores of the detections on the attention regions are reduced after being attacked.

other models. Excluding attacking itself, these adversarial images best attack YOLOv8s and worst attack Swin-T with 91.19% (dropping the model's mAP from 33.26 to 2.93) and 73.61% (from 32.47 down to 8.57) success attack rates, respectively.

6.2 Cross-Domain Datasets Validation

To verify the generality of our attacking method, we also conduct experiments in which models are trained on one dataset and evaluated on another dataset. As presented in Sec.6.1, models are trained on MS COCO 2017, and adversarial examples are also generated from MS COCO 2017. Tab.5 shows that transferability is robust on another dataset, where pre-trained models on MS COCO 2017 are tested with adversarial examples generated from the validation set of PASCAL VOC 2012.

Similar to Tab.4, our method again shows its best performance when testing its adversarial examples (against YOLOv8x) on other models, where these adversarial examples best attack YOLOv8n and worst attack Swin-T with 99.31% (from 45.15 down to 0.31) and 73.61% (from 53.35 down to 0.67) success attack rates, respectively. Moreover, the generated adversarial examples against YOLOv8x on the PASCAL VOC 2012 validation set even outperform those generated on the MS COCO 2017 validation set; indeed, they achieve the average success attack rates of 99% compared to 86.6% of average success attack rate.

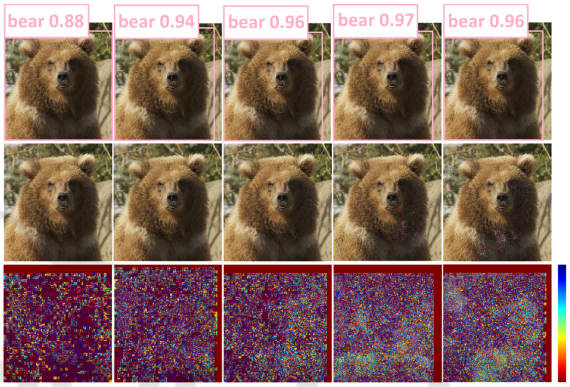

Figure 9: Comparisons between **added distortion amounts (bottom row) on bounding box regions** to fool YOLOv8 from the smallest to the largest size, respectively (by column). Similarity scores computed by Eq.11 between original and perturbed images are 0.9996, 0.9962, 0.9925, 0.9580, and 0.9436, respectively.

Table 3: Success attack rates between DAG (Xie et al., 2017), UEA (Wei et al., 2019), and our method on **one-stage** and **two-stage** detection algorithms.

	One-Stage			Two-Stage		
Baseline	68.00	68.00	25.04	70.10	70.10	27.90
DAG (Xie et al., 2017)	5.00	-	-	64.00	-	-
UEA (Wei et al., 2019)	-	5.00	-	-	20.00	-
Ours	-	-	1.69	-	-	2.10
Succ. Rate	92.65%	92.65%	93.25%	8.70%	71.47%	92.47%

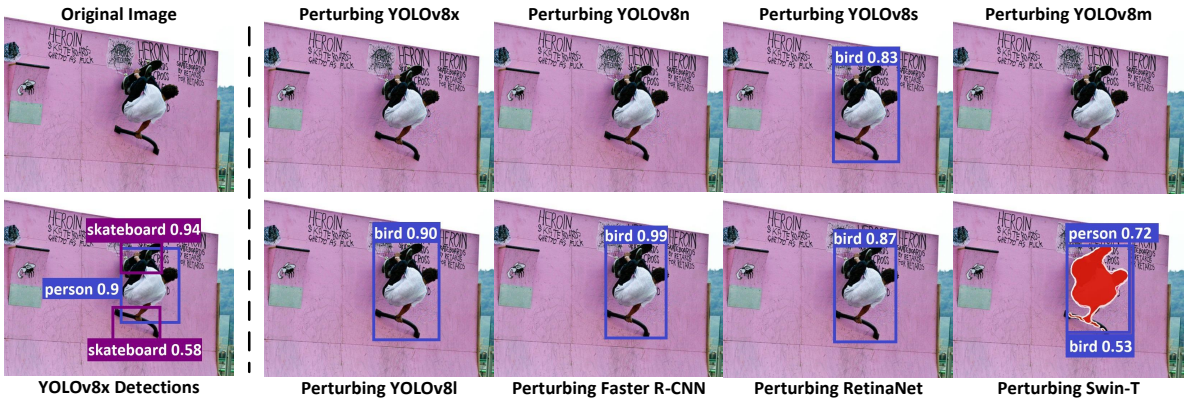

Figure 10: Qualitative results of adversarial images against YOLOv8x that perturbs other detection models, including YOLO's versions, Faster R-CNN, RetinaNet, and Swin Transformer, at confidence thresholds of 0.50. The image is taken from the MS COCO 2017 dataset.

Table 4: **Cross-model transferability among commonly used detection models (in mAP)** of various-sized YOLO's, Faster R-CNN, RetinaNet, and Swin Transformer, at confidence thresholds of 0.50. Each model is evaluated on **the MS COCO 2017 validation set** as a baseline. Meanwhile, Alg.1 best performs attacks on other models when generating adversarial perturbation against YOLOv8x.

Added Perturbation	YOLOv8n	YOLOv8s	YOLOv8m	YOLOv8l	YOLOv8x	Faster R-CNN	RetinaNet	Swin-T
None (baseline)	25.04	33.26	36.98	38.94	40.02	27.90	22.90	32.47
YOLOv8n	0.06	18.12	25.19	28.25	29.52	13.57	10.69	17.22
YOLOv8s	3.32	0.03	16.71	20.68	22.45	9.68	7.31	13.66
YOLOv8m	2.21	4.35	0.02	13.12	15.32	7.03	5.01	10.69
YOLOv8l	1.69	3.52	6.90	0.02	11.37	6.36	4.35	10.18
YOLOv8x	1.42	2.93	5.47	6.50	0.05	5.32	3.58	8.57
Faster R-CNN	3.86	6.96	10.51	13.09	13.96	0.10	0.60	12.70
RetinaNet	6.01	9.99	14.22	17.01	18.01	2.10	0.30	16.00
Swin-T	2.98	5.83	9.49	12.42	14.50	11.30	8.70	0.10

Table 5: **Cross-model transferability among commonly used detection models (in mAP)** of various-sized YOLO's, Faster R-CNN, RetinaNet, and Swin Transformer, with confidence thresholds set to 0.50. Each model is evaluated on **the PASCAL VOC 2012 validation set** as a baseline. Again, Alg.1 best performs attacks on other models when generating adversarial perturbation against YOLOv8x.

Added Perturbation	YOLOv8n	YOLOv8s	YOLOv8m	YOLOv8l	YOLOv8x	Faster R-CNN	RetinaNet	Swin-T
None (baseline)	45.15	54.45	60.80	63.47	64.00	46.13	49.54	53.35
YOLOv8n	0.34	0.64	0.92	1.23	1.25	0.65	0.89	1.03
YOLOv8s	0.36	0.39	0.80	1.07	1.08	0.60	0.86	0.95
YOLOv8m	0.34	0.43	0.52	0.90	1.00	0.58	0.78	0.87
YOLOv8l	0.35	0.48	0.65	0.70	0.88	0.49	0.62	0.75
YOLOv8x	0.31	0.45	0.61	0.66	0.72	0.41	0.58	0.67
Faster R-CNN	5.13	9.04	16.02	18.51	19.75	0.09	1.42	17.23
RetinaNet	8.84	13.94	21.47	23.89	25.57	1.97	0.12	21.97
Swin-T	2.99	6.06	12.39	15.30	18.18	12.18	17.38	0.18

6.3 Transferability to Different Backbones

Furthermore, we compare our methods with DAG (Xie et al., 2017) regarding the transferability to other backbones: the adversarial images generated against a different backbone are used to attack detectors with ResNet-50 as backbones. In specific, we used the images (from the PASCAL VOC dataset) generated against YOLOv8x to perturb Faster R-CNN, RetinaNet, and Swin Transformer. As shown in Tab.2, we can still achieve a success attack rate of 80.44%, 84.37%, and 73.61%, respectively; meanwhile, DAG only achieved 16.23% while performing the same task.

6.4 Consistency with Detection Algorithms

Also, to see how consistent Alg.1 performs with different detection algorithms, we experiment it on both one-stage and two-stage detection algorithms and compare our results with DAG (Xie et al., 2017)

and UEA (Wei et al., 2019), as depicted in Tab.3. All three methods provide high results (above 90%) on one-stage detection methods; however, the performances of DAG and UEA drop when performing adversarial attacks on two-stage detection methods, while our proposed technique can still maintain a consistent success attack rate of 92.47% compared to 93.25% from one-stage methods.

6.5 Qualitative Results

From Tab.4 and Tab.5, we conclude that adversarial images generated against YOLOv8x maintain the best overall transferability and consistency of attacks to other models. As shown in Fig.10, the qualitative results of a perturbed image against YOLOv8x can make other detection models misdetect. Fig.10 also shows that the perturbation amount is imperceptible, the stable transferability to other backbones, and the consistency with one-stage and two-stage methods, restating our key properties in Tab.1.

6.6 Discussions

Our cross-model validation experiments demonstrate the strong transferability of adversarial examples across diverse detection architectures. Adversarial images crafted against YOLOv8x effectively misled other YOLOv8 variants, as well as models like Faster R-CNN, RetinaNet, and Swin Transformer, achieving high success rates. Notably, larger models, such as YOLOv8x, not only demonstrated greater robustness but also generated adversarial examples that generalized better to other models. This trend suggests that larger models architectural complexity enables them to produce perturbations that impact shared features across different backbones.

Cross-domain validations further support the generalizability of our method. Adversarial examples generated on the MS COCO 2017 dataset remained effective when tested on PASCAL VOC 2012, achieving success rates comparable to in-domain experiments. These results underline the robustness of our perturbation approach, which leverages model-agnostic loss gradients to craft transferable adversarial examples. This ability to maintain high efficacy across datasets enhances the practicality of our method for black-box attack scenarios, where access to target model specifics is limited.

The transferability of adversarial examples to different backbones also highlights the adaptability of our approach. Using adversarial examples generated against YOLOv8x, we observed consistent attack success rates on models with ResNet-50 backbones, such as Faster R-CNN and RetinaNet, and even on transformer-based models like Swin Transformer. These findings indicate that our method effectively exploits fundamental vulnerabilities in object detection pipelines, regardless of the underlying network architecture.

Our experiments also confirm the consistency of our method across one-stage and two-stage detection algorithms. While prior methods like DAG and UEA showed a drop in performance on two-stage detectors, our technique maintained high success rates across both categories. This consistency is attributed to the iterative perturbation approach, which accurately targets bounding box regions while controlling distortion, ensuring applicability across different detection paradigms.

Qualitative results and visual analyses provide further evidence of our methods efficacy. Grad-CAM visualizations reveal how adversarial perturbations alter model attention, reducing confidence scores for objects in bounding boxes and eventually leading to misdetections. Additionally, the perturbations remain imperceptible to human observers, striking an effective balance between visual fidelity and attack performance. These properties make our approach suitable for real-world applications where stealth is essential.

Despite these strengths, our method encounters challenges in scenarios involving overlapping bounding boxes, which require more iterations and greater distortion to achieve similar success rates. Addressing these limitations through advanced perturbation strategies or adaptive adversarial training could enhance the robustness of future detection systems. Furthermore, exploring domain adaptation techniques may improve cross-domain transferability even further.

7 CONCLUSIONS

This paper presents a distortion-aware adversarial attack technique on bounding boxes of state-of-the-art object detectors by leveraging target-attacked pixel gradient ascents. By knowing the gradient ascents of those pixels, we iteratively add the perturbation amount to the original image's masked regions until the success attack rate or distortion threshold is obtained or until the detector no longer recognizes the presented objects. To verify the effectiveness of the proposed method, we evaluate our approach on MS COCO 2017 and PASCAL VOC 2012 datasets and achieve success attack rates of up to 100% and 98%, respectively. Also, through validating cross-model transferability, we prove that our method can perform

black-box attacks when generating primary adversarial images on YOLOv8x. As the original motivation of our work, we propose this method to expose the vulnerabilities in neural networks and facilitate building more reliable detection models under adversary attacks. However, we reserve the task of improving the model's robustness for future works. Upon social goods, we also make our source code available to encourage others to build defense methods for this attack method.

REFERENCES

Alaifari, R., Alberti, G. S., and Gauksson, T. (2018). Adef: an iterative algorithm to construct adversarial deformations. In *International Conference on Learning Representations*.

Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection.

Carlini, N. and Wagner, D. (2017). Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee.

Chen, P.-Y., Sharma, Y., Zhang, H., Yi, J., and Hsieh, C.-J. (2018). Ead: elastic-net attacks to deep neural networks via adversarial examples. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Dang, T., Nguyen, K., and Huber, M. (2023). Multiplanar self-calibration for mobile cobot 3d object manipulation using 2d detectors and depth estimation. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1782–1788. IEEE.

Dang, T., Nguyen, K., and Huber, M. (2024). V3d-slam: Robust rgb-d slam in dynamic environments with 3d semantic geometry voting. *arXiv preprint arXiv:2410.12068*.

Du, A., Chen, B., Chin, T.-J., Law, Y. W., Sasdelli, M., Rajasegaran, R., and Campbell, D. (2022). Physical Adversarial Attacks on an Aerial Imagery Object Detector. pages 1796–1806.

Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111:98–136.

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2015). Explaining and harnessing adversarial examples. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Im Choi, J. and Tian, Q. (2022). Adversarial attack and defense of yolo detectors in autonomous driving scenarios. In *2022 IEEE Intelligent Vehicles Symposium (IV)*, pages 1011–1017. IEEE.

Jocher, G., Chaurasia, A., and Qiu, J. (2023). YOLO by Ultralytics.

Kurakin, A., Goodfellow, I. J., and Bengio, S. (2018). Adversarial examples in the physical world. In *Artificial intelligence safety and security*, pages 99–112. Chapman and Hall/CRC.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer.

Lindeberg, T. (2012). Scale invariant feature transform.

Liu, X., Yang, H., Liu, Z., Song, L., Chen, Y., and Li, H. (2019). DPATCH: an adversarial patch attack on object detectors. In Espinoza, H., hÉigeartaigh, S. Ó., Huang, X., Hernández-Orallo, J., and Castillo-Effen, M., editors, *Workshop on Artificial Intelligence Safety 2019 co-located with the Thirty-Third AAAI Conference on Artificial Intelligence 2019 (AAAI-19), Honolulu, Hawaii, January 27, 2019*, volume 2301 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022.

Lu, J., Sibai, H., and Fabry, E. (2017). Adversarial Examples that Fool Detectors. arXiv:1712.02494 [cs].

Lu, Y. (2019). The Level Weighted Structural Similarity Loss: A Step Away from MSE. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):9989–9990. Number: 01.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Moosavi-Dezfooli, S.-M., Fawzi, A., and Frossard, P. (2016). DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks. pages 2574–2582.

Nguyen, K., Dang, T., and Huber, M. (2024a). Real-time 3d semantic scene perception for egocentric robots with binocular vision. *arXiv preprint arXiv:2402.11872*.

Nguyen, K., Dang, T., and Huber, M. (2024b). Volumetric mapping with panoptic refinement using kernel density estimation for mobile robots.

Puccetti, T., Zoppi, T., and Ceccarelli, A. (2023). On the efficacy of metrics to describe adversarial attacks. *arXiv preprint arXiv:2301.13028*.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). Orb: An efficient alternative to sift or surf.

In *2011 International conference on computer vision*, pages 2564–2571. Ieee.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626.

Song, D., Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Tramer, F., Prakash, A., and Kohno, T. (2018). Physical adversarial examples for object detectors. In *12th USENIX workshop on offensive technologies (WOOT 18)*.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612.

Wei, X., Liang, S., Chen, N., and Cao, X. (2019). Transferable adversarial attacks for image and video object detection. In Kraus, S., editor, *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 954–960. ijcai.org.

Xie, C., Wang, J., Zhang, Z., Zhou, Y., Xie, L., and Yuille, A. (2017). Adversarial examples for semantic segmentation and object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1369–1378.