A Survey and Evaluation of Adversarial Attacks in Object Detection

Khoi Nguyen Tiet Nguyen*, Wenyu Zhang*, Kangkang Lu, Yu-Huan Wu, Xingjian Zheng, Hui Li Tan, Liangli Zhen

***Abstract*—Deep learning models achieve remarkable accuracy in computer vision tasks, yet remain vulnerable to adversarial examples–carefully crafted perturbations to input images that can deceive these models into making confident but incorrect predictions. This vulnerability pose significant risks in high-stakes applications such as autonomous vehicles, security surveillance, and safety-critical inspection systems. While the existing literature extensively covers adversarial attacks in image classification, comprehensive analyses of such attacks on object detection systems remain limited. This paper presents a novel taxonomic framework for categorizing adversarial attacks specific to object detection architectures, synthesizes existing robustness metrics, and provides a comprehensive empirical evaluation of state-of-the-art attack methodologies on popular object detection models, including both traditional detectors and modern detectors with vision-language pretraining. Through rigorous analysis of open-source attack implementations and their effectiveness across diverse detection architectures, we derive key insights into attack characteristics. Furthermore, we delineate critical research gaps and emerging challenges to guide future investigations in securing object detection systems against adversarial threats. Our findings establish a foundation for developing more robust detection models while highlighting the urgent need for standardized evaluation protocols in this rapidly evolving domain.**

***Index Terms*—Adversarial Attacks, Adversarial Robustness, Object Detection**

I. Introduction

Deep learning models have demonstrated great success in computer vision on a diverse range of tasks such as image classification, object detection, segmentation, pose estimation, and image captioning [1]–[3]. As these models are increasingly deployed in real-world systems, it is essential to understand the model vulnerabilities that can compromise the safety and security of the systems. *Adversarial examples* are carefully crafted modifications to input data that cause deep learning models to make incorrect predictions, even when these modifications are so subtle they remain imperceptible to humans [4]–[7]. These malicious inputs pose significant safety and security concerns, particularly when deployed against high-stakes applications like autonomous vehicles, security surveillance, and safety-critical inspection systems.

TABLE I: List of notations and acronyms used in our paper. Detailed definitions are provided in the respective sections. Refer to Table III for acronyms of attack methods.

Notations & Acronyms	Brief Definition
x	Clean input image
δ	Perturbation to inject to the clean image x
x'	Adversarial image defined by $x' = x + \delta$
δ^*	The optimal perturbation obtained after some iterative updates
y	True label for x containing J objects, $y[j] = (b_x, b_y, b_h, b_w, c), j \in J$
b_x, b_y, b_h, b_w	b_x, b_y are coordinates of top-left point, b_h, b_w are height and width, of object bounding box b
c	Class label of the object
$\hat{y}$	Object detector output for x
y'	Target label for x'
$\tilde{L}_{od}(x, y, \Theta)$	Training objective of object detection model parameterized by Θ, combining classification loss $\tilde{L}_{clf}$, objectness loss $\tilde{L}_{obj}$, localization regression loss $\tilde{L}_{reg}$
ϵ	Perturbation budget; Threshold value for perturbation constraint such that $d(x', x) < \epsilon$ with distance function d
L_p-norm (L_1, L_2, L_∞)	Perturbation norm measuring the amount of perturbation added to the image
IoU	Intersection-over-Union
AP@γ	Average Precision, the precision averaged across all object classes at a fixed IoU threshold γ
mAP	Mean Average Precision, the average of AP@γ across different IoU thresholds
ASR	Attack Success Rate
FPR, FNR	False Positive Rate, False Negative Rate
PSNR	Peak Signal-to-Noise Ratio
SSIM	Structural Similarity Index Measure
RPN	Region Proposal Network
NMS	Non-Maximum Suppression
RoI	Region of Interest potentially containing objects

* These two authors contributed equally to this work.

This research is supported by the Agency for Science, Technology and Research (A*STAR) under the Singapore Aerospace Programme (Grant No. M2215a0067) and the National Research Foundation Singapore and DSO National Laboratories under the AI Singapore Programme (AISG Award No: AISG2-GC-2023-007). (Corresponding authors: Khoi Nguyen Tiet Nguyen, Wenyu Zhang, and Liangli Zhen)

K. N. T. Nguyen was with the Institute for Infocomm Research, A*STAR, Singapore when he completed this work. Email: nguyentiet-nguyenkhoi@gmail.com

W. Zhang, K. Lu, H. L. Tan are with the Institute for Infocomm Research, A*STAR, Singapore. Email: {zhang_wenyu, lu_kangkang, hltan}@i2r.a-star.edu.sg

Y.-H. Wu, X. Zheng, L. Zhen are with the Institute of High Performance Computing, A*STAR, Singapore. Email: {wu_yuhuan, zheng_xingjian, zhenll}@ihpc.a-star.edu.sg

While the literature on adversarial attacks in image classification is extensive, with numerous comprehensive surveys [4]–[9] and evaluations [10]–[12], research examining adversarial attacks on object detection remains relatively limited. This gap is particularly significant given that object detection models present more complex attack challenges than image classification models, due to their varied network architectures, modules, and sub-processes. Existing surveys on object detection adversarial attacks have notable limitations. For instance, Amirkhani *et al.* focused on object detection for the autonomous vehicle application [13]. Mi *et al.* surveyed adversarial attacks and defences but omitted evaluation procedures and comparative attack assessments [14]. Amongst existing evaluation studies [15], [16], they have not addressed attack transferability in the more realistic black-box system scenarios, and some of them focused on patch attacks only while excluding other non-patch-based attacks [16], [17]. Furthermore, conducting a fair comparative analysis of attack method effectiveness based solely on reported results in published articles presents significant methodological challenges. These challenges stem from substantial variations across studies in multiple key dimensions: choice of detection architectures, dataset selection (full [18], [19], subset [20], [21], or proprietary [15]), evaluation metrics (standard mAP [18], [19] versus custom metrics [20], [21]), attack scope (all classes [18] versus specific targets [17], [22]), and hyper-parameters like varying constraints on allowed query volumes [21], [23], [24].

The key factors that have contributed to the scarcity of evaluation studies on object detection-specific adversarial attacks include: 1) the limited availability of source code for many proposed methods and 2) the incompatibility of dependencies across existing open-source implementations, which prevents unified testing across different detection models. While some evaluation efforts exist, they are limited in scope. For example, Xu *et al.* evaluated six adversarial attacks in terms of their effectiveness, computational cost, number of attack iterations and magnitute of image distortion [15]. Similarly, Hingun *et al.* evaluated how adversarial patches affect the robustness of object detectors on road signs [16]. Both of these studies are conducted in a white-box setting, where attackers have complete access to the detection system. In contrast, Du *et al.* and Yang *et al.* explored attack transferability in the black-box setting [25], [26], where attackers have limited system access. However, their scope was narrow: the first one examined only two object detection-specific attack algorithms, while another one focused on comparing different patch patterns using a single patch attack algorithm. A summary of these evaluations is reported in Table II.

In this work, we address these limitations by providing a comprehensive survey and evaluation of adversarial attacks in object detection. Our main contributions are as follows:

- We propose a novel taxonomy of adversarial attacks in object detection. This taxonomy provides a structured framework for categorizing existing attack methods and characterizing their key properties, enabling researchers to better understand the relationships and distinctions between different approaches.
- We conduct a comprehensive analysis of evaluation methodologies in the field, examining the various metrics, benchmark datasets, and model architectures commonly used to assess attack effectiveness. This analysis highlights both standard practices and potential limitations in current evaluation approaches.
- We perform a systematic evaluation of open-source attack methods and adversarial robustness of popular object detection models, including both traditional detectors and modern detectors with vision-language pretraining. Our systematic evaluation provides unprecedented insights into the relative strengths of different attack strategies and the comparative robustness of detection architectures under adversarial conditions.
- Based on our analysis and experimental results, we identify critical gaps in current research and outline promising directions for future work. These include developing more effective adversarial attacks for modern object detectors, designing robust defense mechanisms specifically for small object detection, preventing multimodal adversarial attacks, and advancing the state of physical adversarial attacks that can reliably fool detectors in real-world scenarios.

II. Taxonomy of adversarial attacks

We present our taxonomy of adversarial attacks in object detection in Figure 1. We describe preliminary information in Section II-A, followed by detailed categorizations for adversarial model and attack method in Section II-B and II-C, respectively. We note that some taxonomy categories, namely environment, knowledge of adversary, intent specificity, perturbation norm, attack frequency and attack specificity, are common concepts that are also used in other tasks including image classification. We discuss these taxonomy categories and provide the relevant references specifically in the context of object detection. For better readability, we provide a list of notations and acronyms used in our paper in Table I.

A. Preliminaries

The aim of object detection is to detect, localize and classify objects of interest in an image. Object detectors with real-time inference speed can also be used for streaming video. Denote $(\mathcal{X}, \mathcal{Y})$ as the input-output space, and $x \in \mathcal{X}$ is an input image, and $y \in \mathcal{Y}$ is the output. For an image containing J objects, $y \in \mathbb{R}^{J\times 5}$ with $y[j] = (b_x, b_y, b_h, b_w, c)$ where (b_x, b_y) are the top-left corner coordinates and (b_h, b_w) are the height and width of the object bounding box b, and c is the label out of C object classes. For an object detection model f parameterized by Θ, its prediction for x is $\hat{y} = f(x; \Theta)$. Some models output (b_x, b_y) as the center coordinates, and output additional information such as the confidence score $s = (s_1, \ldots, s_C)$ where s_c is the predictive probability for object class c, and objectness score p of whether an object exists in the predicted box.

The training objective $\tilde{L}_{od}(x, y, \Theta)$ for object detection is typically the weighted combination of the following components for a predicted box [27], [28]:

TABLE II: Evaluation of adversarial robustness in object detection tasks. Yang et al. [26] evaluated one patch attack algorithm across five patterns[a], while Du et al. [25] examined 13 attacks, though only two specifically target object detection rather than image classification[b].

Article	Software		Attack Generation		Robustness Evaluation			
	Framework	Open-source	Knowledge	Norm	# Detectors	# Attacks	Metric	Datasets
[15]	Not specified	N	White	L_2	2 one-stage 1 two-stage	3 3	mAP, # iterations, time cost, distortion	VOC, proprietary video
[16]	PyTorch	Y	White	L_2	1 one-stage 1 two-stage	1 1	ASR, FNR	Mapillary, Traffic Sign
[25]	Tensorflow	N	White/Black	L_p	6 one-stage 12 two-stage	2+11[b] 2+11[b]	mAP	VOC, COCO
[26]	PyTorch	N	White/Black	-	2 one-stage 2 two-stage	1[a] 1[a]	mAP	VOC, COCO, Inria Person

- Classification loss $\tilde{L}_{clf}(x, y; \Theta) = -\sum_{i=1}^{C} s_i \log(\hat{s}_i)$. Besides binary cross entropy, other choices include multi-class cross entropy and focal loss.
- Objectness loss $\tilde{L}_{obj}(x, y; \Theta) = -p \log(\hat{p}) - (1 - p) \log(1 - \hat{p})$.
- Localization regression loss $\tilde{L}_{reg}(x, y; \Theta) = (b_x - \hat{b}_x)^2 + (b_y - \hat{b}_y)^2 + (b_h - \hat{b}_h)^2 + (b_w - \hat{b}_w)^2$. Besides squared error, other choices include L1 and smooth L1 losses.

An adversarial attack is the addition of perturbation δ to image x such that model prediction on the perturbed image $x' = x + \delta$ is incorrect *i.e.*, $f(x'; \Theta) = f(x + \delta; \Theta) \neq y$. Denoting the adversarial loss as $L_{adv}(x+\delta, y; \Theta)$ with f as the target model, the attacker search for the optimal perturbation δ^* by

$$\delta^* = \arg\min_{\delta} L_{adv}(x + \delta, y; \Theta). \tag{1}$$

For controlling the amount of perturbation added, a perturbation constraint can be added *i.e.*, $d(x+\delta, x) < \epsilon$ with distance function $d(a, b)$ and threshold $\epsilon > 0$.

B. Adversarial model

1) Environment: **Digital:** Digital attacks alter the pixels of images input into the target model. The attacker has no access to the physical environment and the image capture and pre-processing procedures. We focus on digital attacks in this literature review, as physical attacks are often extensions of digital attacks by additionally considering variations in physical conditions, *e.g.*, lighting, object pose, camera angle.
Physical: While we focus on digital attacks in the literature review, we note that some papers implemented physical versions of their proposed attacks as sticker attacks [26]. A common strategy is to use Expectation Over Transformation (EOT) [26], [29] which augments the input image x with a sampled set of transformations $\{t\}$ from distribution T to simulate real-world transformations such as lighting and viewing angle changes, and optimizes the adversarial image $x' = x + \delta$ over the expectation of the transformed images with the EOT loss:

$$\delta^* = \arg\min_{\delta} \mathbb{E}_{t\sim T} L_{adv}(x + \delta, y; \Theta) \tag{2}$$

subject to $\mathbb{E}_{t\sim T} d(t(x + \delta), t(x)) < \epsilon$ with distance function $d(a, b)$ and threshold ϵ.

Thys *et al.* proposed to minimize a non-printability score to favor pixel values close to a set of printable colors $Color_{print}$ [30]:

$$L_{nps}(x') = \sum_{i,j} \min_{color \in Color_{print}} |x'_{i,j} - color| \tag{3}$$

where $x'_{i,j}$ is the pixel at location (i, j). Non-printability refers to the challenge of generating adversarial attacks that remain effective after printing and affixing on real-world objects, by considering physical constraints such as color gamut limitations, resolution and detail loss, and lighting and other environmental conditions.

Thys *et al.* also considered to minimize a total variation loss to produce images with smooth color transitions instead of noisy pixels which may not be captured well by detectors in physical environments [30] :

$$L_{tv}(x') = \sum_{i,j} \sqrt{(x'_{i,j} - x'_{i+1,j})^2 + (x'_{i,j} - x'_{i,j+1})^2}. \tag{4}$$

The evaluation of attacks and defenses in the real world is exceptionally costly. Hingun *et al.* utilized more realistic simulation to study the effect of real-world patch attacks on road signs, by simulating physical adversarial stickers subject to different locations, orientations and lighting conditions [16]. However, they did not evaluate the fidelity of the simulated images to real-world images.

2) Knowledge: **White-box attacks:** In a white-box attack, the adversary has complete knowledge of the target model. Information includes the model input and output, neural network architecture, parameter weights and gradients, decision-making process, and training procedure and dataset. The attacker either has full access to the target model or has the capability to completely reconstruct the target model. With this level of information, the adversary can craft attacks specific to the target model.
Gray-box attacks: In a gray-box attack, the adversary has some but incomplete knowledge of the target model. For instance, the adversary may not have knowledge of the network architecture or training dataset [31]. For defended models, Yin *et al.* assumed that the adversary knows the defense neural network architecture but not the parameters [32]. In the gray-box setting, the adversary can use a surrogate model similar to the target model to craft attacks instead [31], [32]. Due to

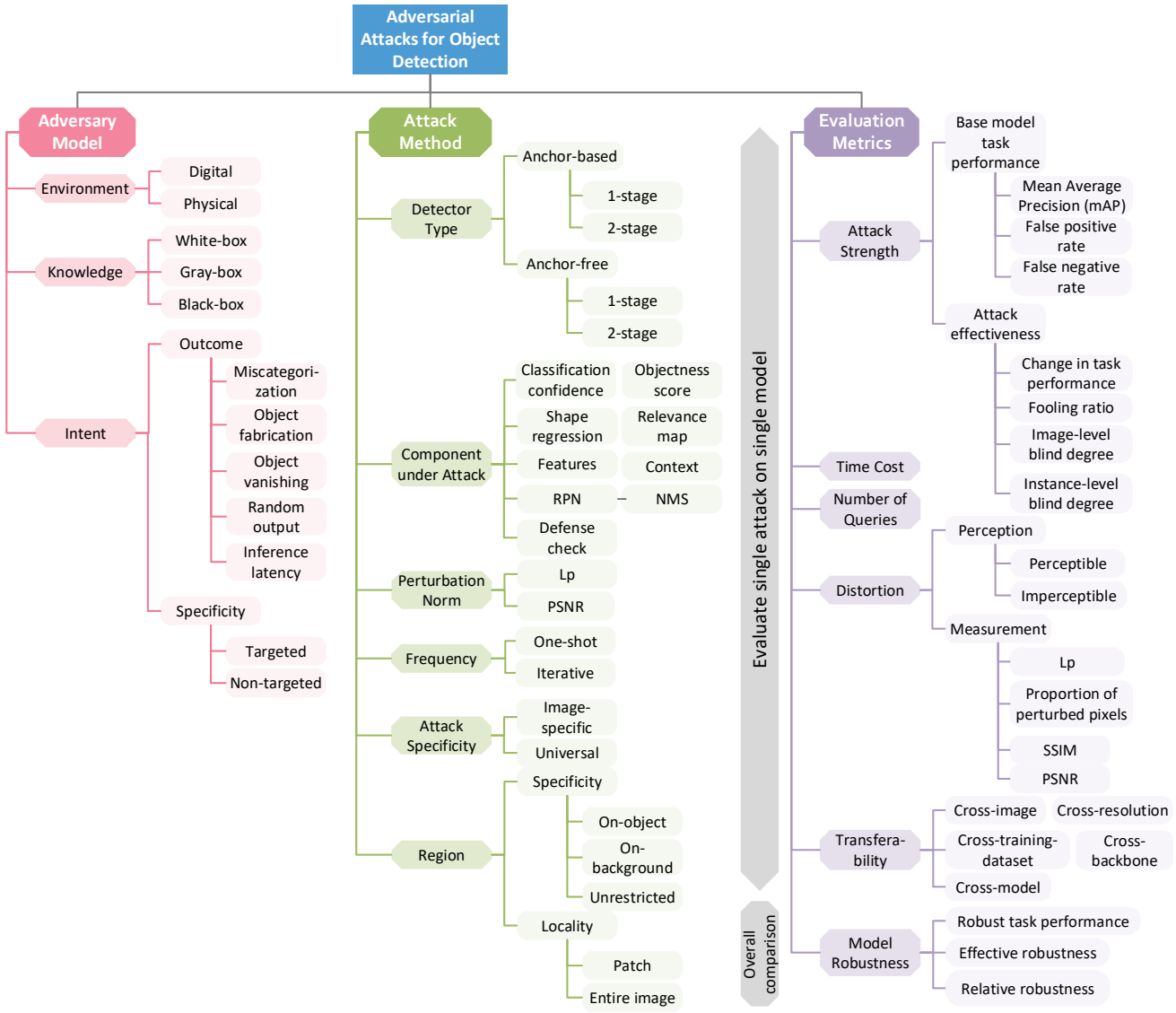

Fig. 1: Taxonomy and evaluation metrics on adversarial attacks in object detection.

the limitation in information available, gray-box attacks are typically less effective than white-box attacks.

Black-box attacks: In a black-box attack, the adversary has no knowledge of the target model. The adversary may submit input queries and observe the predicted bounding boxes and class confidence scores [23]. Common models and algorithms can be used as surrogates as they may share vulnerabilities with the target model [24]. Cai *et al.* utilized an ensemble of M surrogate models and proposed to minimize a joint adversarial loss [33]:

$$L_{adv-ensemble}(x+\delta, y; \{\alpha_m\}, \{\Theta_m\}) = \sum_{m=1}^{M} \alpha_m L_{adv}(x+\delta, y; \Theta_m) \quad (5)$$

where $L_{adv}(x+\delta, y; \Theta_m)$ is the loss for the m-th surrogate model f_m parameterized by Θ_m, $\alpha_m > 0$ for $m \in \{1, \ldots, M\}$ and $\sum_{m=1}^{M} \alpha_m = 1$. Black-box attacks are the most challenging attacks to execute effectively. However, due to the lack of assumption on target model information, they are versatile to implement and are applicable to a wide range of models.

3) Intent outcome: Integrity-based attacks aim to compromise the accuracy of the model predictions [28].

Random output: The adversary perturbs the input such that the model detection output is different from the ground-truth, but does not have a specific intended outcome on the accuracy of the predicted bounding box or object label. An example is to learn δ by maximizing the detection training objective $\tilde{L}_{od}(x+\delta, y; \Theta)$.

Object vanishing: The goal of the adversary is for the model to miss detecting objects in the input image. For examples, [26] and [30] attack such that no human is detected in the inputs. In [19], bounding boxes can be shrunk in size. The adversary can learn δ to reduce objectness scores of candidate boxes.

Object fabrication: The goal of the adversary is for the model to produce redundant bounding boxes or to detect non-existent objects in the input image. The adversary can learn δ

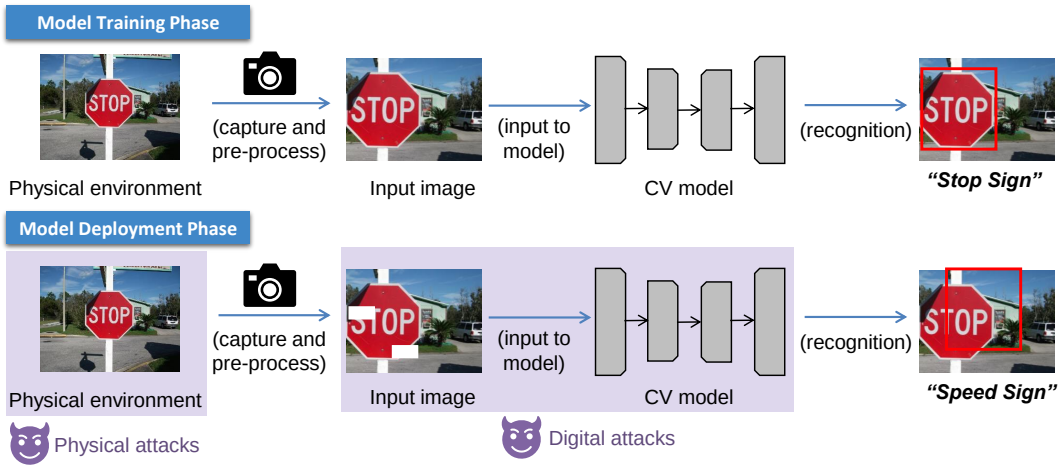

Fig. 2: Adversarial attack procedure.

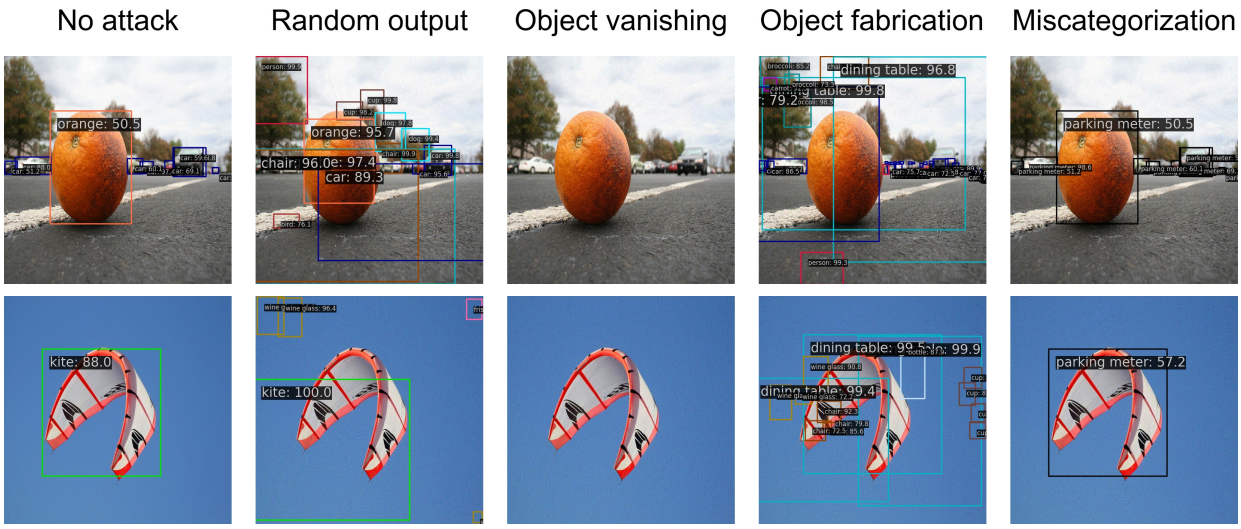

Fig. 3: Examples of outcomes of integrity-based attacks. Original images taken from MS COCO 2017 [34]

to increase objectness scores of candidate boxes.

Miscategorization: The goal of the adversary is to perturb the input such that the model predicts an incorrect label for a detected object. Using the training objective for object detection as the adversarial loss, the adversary can learn δ by minimizing $\tilde{L}_{od}(x+\delta, y'; \Theta)$ with target class label $y' \neq c$.

Availability-based attacks aim to compromise uninterrupted access to the model to process inputs.

Inference latency: The goal of the adversary is to increase the model's inference time, which can disrupt real-time applications. [35] fabricate boxes for non-existent objects to overload the Non-Maximum Suppression (NMS) [36] algorithm.

4) Intent specificity: **Targeted:** In targeted attacks, the adversary perturbs the input such that for the attacked object, the model assigns it the class label specified by the adversary. In object vanishing attacks, the target is the background class or the non-detection of objects existent in the image.

Non-targeted: In non-targeted attacks, the adversary do not specify class labels to be predicted for attacked objects.

C. Attack method

1) Detector type: We categorize object detectors by their usage of anchor boxes and the number of stages in the algorithm.

Anchor-based techniques are commonly used in deep learning object detection algorithms. Anchor-based detectors create anchor boxes of various shapes and sizes placed at different locations on the image to represent the objects to be detected. The anchor boxes act as priors and are refined as the object detection network learns to predict the objectness of the anchor boxes, offset of the anchor boxes from the ground-truth boxes, as well as the object class labels. Redundant boxes are removed to output the final predicted bounding boxes.

One-stage anchor-based: One-stage anchor-based detectors directly predicts the bounding box coordinates and class labels of objects in an image. Examples include SSD [37] which use pre-defined anchor boxes, and YOLO v4 [38] which learns anchor boxes during training.

Two-stage anchor-based: In the first stage, a set of candidate

object regions is generated by a region proposal algorithm such as a region proposal network (RPN) that utilizes anchor boxes [39]. In the second stage, a neural network refines and classifies the region proposals. Examples include Faster R-CNN [39] and Libra R-CNN [40].

[41] empirically finds that it is more difficult to attack two-stage detectors as compared to one-stage detectors. Two-stage detectors like Faster R-CNN have features with a smaller receptive field than one-stage detectors like YOLO v4, which contribute to increased robustness to local perturbations.

A disadvantage of anchor-based detectors is the need to process a large number of anchor boxes in order to cover objects of different shapes and sizes, which can increase computation time. Anchor-free methods do not need anchor boxes as priors, and hence can do away with hand-crafted components for anchor generation.

One-stage anchor-free: One-stage anchor-free detectors directly predict the coordinates of the bounding boxes. Examples include YOLO [36], FoveaBox [42] and FCOS [43]. Apart from CNN-based models, transformers-based models such as DETR [44] and Deformable DETR [45] are more recently developed. They further simplified the detection process, directly predicting tokens as bounding boxes, whose process is supervised via Hungarian matching. Therefore, they do not rely on non-maximum suppression that most CNN-based detectors applied on.

Two-stage anchor-free: Similar to two-stage anchor-based methods, a region proposal algorithm generates candidate object regions in the first stage, and the region proposals and refined in the second stage. The region proposal algorithm, such as Selective Search [46], does not make use of anchor boxes. Examples of such detectors include R-CNN [46], Corner Proposal Network [47], and Sparse R-CNN [48].

2) Component under attack: An adversarial method can attack a combination of components to achieve the desired outcome of the adversary.

Classification confidence score: From our summary Table IV, most methods attack the classification confidence score s. Attacks achieve misclassification in detection by increasing the predictive probability of an incorrect label over that of the correct label [18], [31], [41]. UEA [49] uses a Generative Adversarial Network (GAN) to generate perturbations that produce inaccurate confidence scores.

For models that do not estimate a separate objectness score, object class confidence is used as objectness score. For object vanishing attacks, object class confidence is lowered [23], [26], [50] and background score is raised [51]–[53]. For object fabrication attacks, object class confidence is increased [35].

Objectness score: The objectness scores of predicted boxes can be increased for object fabrication [18] or decreased for object vanishing [18], [22], [30] attacks. TransPatch [22] utilizes a transformer generator to create perturbations.

Shape regression: By attacking the shape regression module, predicted box location and size becomes inaccurate [18], [52], [54], [55]. PRFA [23] reduce the Intersection-over-Union (IoU) of predicted and ground-truth boxes. Daedalus [24] and [35] compress dimensions of predicted boxes to create more redundant boxes.

Relevance map: Authors for RAD [19] observes that relevance maps from detection interpreters is common across detectors and such that a wider range of detectors are vulnerable to relevance map attacks in black-box setting.

Features: UEA [49] regularizes features of foreground objects to be random values to damage object information in the features.

Context: CAP [56] damages contextual information in an identified Region of Interest (RoI) by decreasing object classification scores and increasing background scores of the contextual region. Based on the co-occurrence object relation graph of the victim object, [33] identifies helper objects that typically co-occur with the victim object, add perturbations to modify labels of both the victim and helper objects to increase attack success rate.

Region proposal algorithm: In two-stage detectors, when quality of proposals from the region proposal algorithm in the first stage is degraded, consequent detection performance in the second stage will be affected. G-UAP [53] decreases confidence score of foreground and increases confidence score of background so that the region proposal network (RPN) mistakes foreground for background. R-AP [55] degrades RPN performance by reducing objectness score and disturbing shape regression.

Non-Maximum Suppression: Non-Maximum Suppression (NMS) [36] is a post-processing technique to remove redundant bounding boxes generated by the object detector. For a given object class, the detected bounding box with the highest confidence score above a specified threshold is selected, and other bounding boxes with overlap above a specified IoU threshold are discarded. Figure 4 shows the bounding boxes before and after NMS post-processing in YOLO [36].

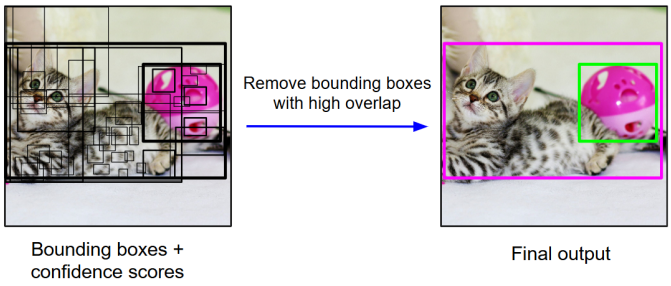

Fig. 4: Non-Maximum Suppression (NMS) a post-processing technique to remove redundant bounding boxes generated by the object detector. Original image taken from MS COCO 2017 [34].

Daedalus [24] fabricate non-existent objects by reducing the number of bounding boxes filtered off by NMS. Perturbations are added to maximize confidence scores of the detected bounding boxes, and to minimize IoU for each pair of boxes. [35] seeks to increase inference time by increasing detected bounding boxes to load the NMS module.

Defence check: Context consistency checks aim to detect attacks through inconsistent object-to-object co-occurrence relationships that can arise in an attacked image. ADC [32] and ZQA [21] can evade context consistency checks by

assigning class labels consistent with normal co-occurrence relationships.

3) Perturbation norm: A perturbation constraint can be added to the adversarial loss to control the amount of perturbation added, such that the perturbations are visually imperceptible to human observers. Typically, the L_p-norm, $\|\delta\|_p$ is used, with $p \in \{0, 1, 2, \infty\}$. CAP [56] uses the Peak Signal-to-Noise Ratio (PSNR):

$$PSNR = 10 \log_{10} \frac{MAX(x)^2}{MSE(x, x+\delta)} \tag{6}$$

where $MAX(x)$ is the maximum pixel value in x, and $MSE(x, x+\delta) = \frac{1}{h(x)w(x)} \|\delta\|_2^2$ for image x with height $h(x)$ and width $w(x)$.

4) Frequency: **One-shot:** In a one-shot attack, the adversary has a single attempt to perturb the input image. One-shot attacks may be less effective than iterative attacks, but are typically more time and resource-efficient, and consequently more suitable for real-time attacks. Such perturbations can be generated by Generative Adversarial Networks (GANs) [49]. ZQA [21] estimates a perturbation success probability matrix to select the list of victim objects and labels that is most likely to lead to a successful attack, and does not iteratively query the target model to check the outcome of the attack.

Iterative: In an iterative attack, the adversary can repeatedly optimize the perturbation based on feedback from the target model. Many existing methods use iterative gradient updates, such as projected gradient descent (PGD) [57], to generate a successful attack based on target model gradients. Black-box methods query the target model to check attack success and to modify the attack plan corresponding to the model output [20], [33]. Iterative attacks typically use more time and resources than one-shot attacks.

5) Attack specificity: **Image-specific:** A separate perturbation needs to be learned for each input image for image-specific attacks. These attacks are generally more effective than universal attacks.

Universal: An universal attack is a perturbation learned to be applicable for any input image. While more challenging to construct than image-specific attacks, effective universal attacks pose a more significant threat as they are transferable across images without further accessing the target model. Universal perturbations are learned using a training set of images typically similarly distributed to the test images [51]. Additional image-specific finetuning on universal perturbations can be applied to improve their effectiveness [53].

6) Region specificity: **On-object:** Some attacks focus their perturbations on pixels of foreground objects to cause miscategorization and object vanishing. The objects can be manually located [26], or objectness of pixels can be estimated through detector features or outputs [22], [23], [49]. On-object perturbations can be constructed into stickers, posters, clothes, or other surfaces to be affixed to the object for physical attacks [26].

On-background: Some methods focus their perturbations on the background. For example, Li *et al.* increased the number of false positives by perturbing the background [52].

Unrestricted: Without placing a restriction on the type of pixels under attack, the entire image can be perturbed.

7) Region locality: **Patch:** Patch attacks constrain perturbations within region of a fixed shape, typically a rectangle. Some methods allow multiple patches [23], [50]. Patches of pre-specified sizes can be constructed into stickers for physical attacks [22], [26], [30].

Entire image: Without constraining perturbations to pixels within a patch, the entire image can be perturbed. A perturbation norm on δ is typically used to constraint the magnitude of perturbation.

We summarize a list of articles on adversarial attack methods in object detection in Table III and IV. We base the knowledge of the adversary model on the method description as well as the transferability experiments, that is, we include gray and/or black-box attack capabilities for a white-box attack if the article demonstrates that the attack has high transferability despite limited knowledge of the target model. We include gray-box capability for cross-resolution, cross-training-dataset or cross-backbone transferability, and black-box capability for cross-model transferability. Description on transferability types can be found in Section III-E.

III. Evaluation Metrics on Adversarial Attacks and Model Robustness

We summarize evaluations metrics on adversarial attacks and model robustness in Figure 1. We include metrics used in existing works to evaluate attacks on object detection models, as well as propose relevant metrics from other domains, and elaborate on these metrics in Section III-A to III-F. We summarize datasets and object detection models used in existing works in Section III-G.

A. Attack strength

1) Base model task performance: Given a fixed Intersection-over-Union (IoU) threshold, a true positive is a predicted bounding box that has an IoU with a ground-truth box that meets or exceeds the threshold. A true negative is an object that the detector does not detect and is not in the ground-truth set. A false positive is a falsely detected object and a false negative is an object that the detector failed to detect.

False positive rate (FPR): $FPR = \frac{\text{false positives}}{\text{false positives + true negatives}}$ is the ratio of falsely detected objects to the total number of negative examples.

False negative rate (FNR): $FNR = \frac{\text{false negatives}}{\text{false negatives + true positives}}$ is the ratio of missed objects to the total number of objects.

The Average Precision (AP) and Mean Average Precision (mAP) are common metric for evaluating the performance of object detection algorithms. The definitions of the metrics differ in some cases. For clarify, we define the metrics below.

Average Precision (AP@γ): We define AP@γ as the precision averaged across all object classes at a fixed IoU threshold γ.

Mean Average Precision (mAP): We define mAP as the average of AP@γ across T IoU thresholds $\{\gamma_t\}_{t=1}^T$, i.e.

TABLE III: Articles on adversarial attack methods in object detection. * denotes open-source code is available. For detector type, 1-AB, 2-AB, 1-AF, 2-AF denote one-stage anchor-based, two-stage anchor-based, one-stage anchor-free and two-stage anchor-free detectors, respectively.

Article	Adversary Model				Attack Model					
	Environment	Knowledge	Outcome	Intent Specificify	Detector Type	Frequency	Attack Specificity	Perturbation Constraint	Region Specificity	Locality
DAG* [31]	Digital	White/Gray/Black	Miscategorization	Targeted	2-AB	Iterative	Image-specific	-	Unrestricted	Entire image
Li et al. [52]	Digital	White/Gray	Random output	Non-targeted	1-AB/2-AB	Iterative	Image-specific	-	On-background	Patch
Yang et al. [26]	Digital/Physical	White	Object vanishing	Targeted	1-AB	Iterative	Image-specific	-	On-object	Patch
R-AP* [55]	Digital	Black	Random output	Non-targeted	2-AB	Iterative	Image-specific	Peak Signal-to-Noise Ratio	Unrestricted	Entire image
UEA* [49]	Digital	White	Random output	Non-targeted	1-AB/2-AB	One-shot	Image-specific	-	On-object	Entire image
DPatch* [54]	Digital	White/Gray/Black	Random output	Targeted/ Non-targeted	1-AB/2-AB	Iterative	Image-specific	-	Unrestricted	Patch
G-UAP [53]	Digital	White/Gray/Black	Object vanishing	Targeted	2-AB	Iterative	Universal	L_∞	Unrestricted	Entire image
Thys et al. [30]	Digital/Physical	White	Object vanishing	Targeted	1-AB	Iterative	Image-specific	-	On-object	Patch
CAP [56]	Digital	White	Object vanishing	Targeted	2-AB	Iterative	Image-specific	Peak Signal-to-Noise Ratio	Unrestricted	Entire image
TOG* [18]	Digital	White/Gray/Black	Miscategorization/ Object fabrication/ Object vanishing	Targeted	1-AB	Iterative	Image-specific/ Universal	$L_0/L_2/L_\infty$	Unrestricted	Entire image
DPAttack* [41]	Digital	White	Object vanishing	Targeted	1-AB/2-AB	Iterative	Image-specific	-	On-object	Patch
Evaporate Attack [58]	Digital	Black	Object vanishing	Targeted	1-AB/2-AB	Iterative	Image-specific	L_2	Unrestricted	Entire image
U-DOS [51]	Digital	White/Gray/Black	Object vanishing	Targeted	1-AB/2-AB	Iterative	Universal	L_∞	Unrestricted	Entire image
RPAttack* [50]	Digital	White	Object vanishing	Targeted	1-AB/2-AB	Iterative	Image-specific	-	Unrestricted	Patch
PRFA* [23]	Digital	Black	Random output	Non-targeted	1-AB/2-AB/ 1-AF	Iterative	Image-specific	L_p	On-object	Patch
RAD* [19]	Digital	Black	Random output	Non-targeted	1-AB/2-AB	One-shot	Image-specific	L_∞	Unrestricted	Entire image
ADC [32]	Digital	White/Gray	Miscategorization/ Object fabrication/ Object vanishing	Targeted	2-AB	Iterative	Image-specific	L_∞	Unrestricted	Entire image
Daedalus* [24]	Digital	White	Object fabrication	Non-targeted	1-AB	Iterative	Image-specific	L_0/L_2	Unrestricted	Entire image
CAT * [33]	Digital	Black	Miscategorization	Targeted	1-AB/2-AB/ 1-AF	Iterative	Image-specific	L_∞	Unrestricted	Entire image
TransPatch [22]	Digital/Physical	White	Object vanishing	Targeted	1-AB/ 1-AF	Iterative	Image-specific	-	On-object	Patch
ZQA [21]	Digital	Black	Miscategorization	Targeted	1-AB/2-AB/ 1-AF	One-shot	Image-specific	l_∞	Unrestricted	Entire image
EBAD * [20]	Digital	Black	Miscategorization	Targeted	1-AB/2-AB/ 1-AF	Iterative	Image-specific	l_∞	Unrestricted	Entire image
Shapira et al. [35]	Digital	White	Inference latency	Targeted	1-AB	Iterative	Universal	L_2	Unrestricted	Entire image

TABLE IV: Component attacked by adversarial methods for object detection. * denotes open-source code is available.

Article	Component under Attack								
	Confidence Score	Objectness Score	Shape regression	Relevance Map	Features	Context	Region Proposal	NMS	Defence Check
DAG* [31]	✓								
Li et al. [52]	✓		✓						
Yang et al. [26]	✓								
R-AP* [55]		✓	✓				✓		
UEA* [49]	✓				✓				
DPatch* [54]	✓		✓						
G-UAP [53]	✓						✓		
Thys et al. [30]	✓	✓							
CAP [56]	✓					✓	✓		
TOG* [18]	✓	✓	✓						
DPAttack* [41]	✓								
Evaporate Attack [58]	✓								
U-DOS [51]	✓								
RPAttack* [50]	✓								
PRFA* [23]	✓		✓						
RAD* [19]				✓					
ADC [32]	✓				✓				✓
Daedalus* [24]	✓		✓					✓	
CAT * [33]	✓	✓	✓			✓			
TransPatch [22]		✓							
ZQA [21]	✓	✓	✓			✓			✓
EBDA * [20]	✓	✓	✓			✓			
Shapira et al. [35]	✓		✓					✓	

$mAP = \frac{1}{T}\sum_{t=1}^{T} \text{AP@}\gamma_t$. mAP consolidates detection performance at different IoU thresholds, and thus avoids having to set a fixed threshold for evaluation.

2) Attack effectiveness: **Change in task performance**: The magnitude of decrease in mAP from the base level reflects the overall effectiveness of the adversarial attacks. An increase in FPR indicates successful attacks for miscategorization and object fabrication. An increase in FNR indicates successful attacks for miscategorization and object vanishing.

Fooling ratio: Ratio of successfully attacked images. The success of an attack needs to be defined. For instance, for object vanishing attacks, [41] define a successful attack as one which suppresses all bounding boxes in an image.

For attacks aimed at object vanishing, [51] further utilizes image-level and instance-level blind degree to measure the effectiveness of the attacks.

Image-level blind degree: Ratio of images where at least one object is detected with confidence above a specified threshold.

Instance-level blind degree: Average number of objects detected with confidence above a specified threshold in each image.

With similar metrics, Wu *et al.* proposed to compute the ratio between the metrics evaluated on the original and perturbed images [41].

B. Time cost

The frame rate for real-time object detection depends on the application. Detecting fast objects as in traffic monitoring requires at least 10 frames per second (FPS), while detecting slow objects such as monitoring people passing through an area requires 2-3 FPS [59]. Attack methods that take a long time to generate perturbations are not suitable for real-time object detection applications. From the evaluation in [15],

UEA [49] and [30] take at most 0.05s per image and can be suitable for real-time attacks, while DAG [31], Daedalus [24], R-AP [55] and [26] take from 1 to more than 10 minutes.

C. Number of queries

Attack methods may repeatedly query the target model to access parameter gradients for optimization and to check for attack success. For example in [15], DAG [31] and R-AP [55] use between 50 and 100 queries, while Daedalus [24] uses more than 600 queries. When the adversary does not have the ability to query the target model a large number of times, these query-based methods would be unsuitable. Moreover, by querying the target model, the adversary risks being discovered.

D. Perturbation distortion

1) Perception: An adversarial attack can be categorized as either perceptible or imperceptible based on whether the perturbation can be easily perceived by humans. In general, patch attacks are more easily perceived, while pixel perturbations regularized by a norm constraint are more difficult to perceive.

2) Measurement: The amount of distortion on the original image can be quantitatively assessed. Common measurements are L_p-norm with $p \in \{0, 1, 2, \infty\}$, proportion of perturbed pixels, Peak Signal-to-Noise Ratio (PSNR) as in Equation 6, and Structural Similarity Index Measure (SSIM) comparing the luminance, contrast and structure between two images [60].

$$SSIM(x, x') = \\ luminance(x, x')^{\alpha} \cdot contract(x, x')^{\beta} \cdot structure(x, x')^{\gamma} \tag{7}$$

$$luminance(x, x') = \frac{2\mu_x \mu_{x'} + k_l}{\mu_x^2 + \mu_{x'}^2 + k_l}$$
$$contrast(x, x') = \frac{2\sigma_x \sigma_{x'} + k_c}{\sigma_x^2 + \sigma_{x'}^2 + k_c}$$
$$structure(x, x') = \frac{\sigma_{x,x'} + k_s}{\sigma_x \sigma_{x'} + k_s}$$

with constants k_l, k_c, k_s. The terms of μ_x and σ_x^2 are the pixel mean and variance of x (and similarly defined for x'), and $\sigma_{x,x'}$ is the covariance of x and x'.

E. Transferability

Attack transferability is evaluated by computing the adversarial perturbations for an input image on a training model, and applying the perturbations to attack a different image or target model. Attacks that demonstrate high transferability to target models which differ from the training model can be suitable for gray-box and black-box settings.

Cross-image: Perturbation computed on an input image is applied on a different image.

Cross-resolution: Target model processes images at a different resolution.

Cross-training-dataset: Target model is trained with a different dataset.

Cross-backbone: Target model has a different network architecture. Other aspects of the target model are assumed to be the same as those of the training model.

Cross-model: Target model is a different algorithm from the training model. For instance, training and target models are both one-stage detectors but different algorithms (e.g. SSD $\rightarrow$ YOLO), or training model is a one-stage detector and target model is a two-stage detector (e.g. SSD $\rightarrow$ Faster R-CNN). Cross-model transferability can be difficult to achieve as models may be trained differently and hence use different features for predictions. Learning the perturbation on an ensemble of models have been found to increase attack effectiveness on the unseen target model [20], [24], [35].

F. Model Robustness

Let $perf_{clean}(f)$ and $perf_{adv}(f)$ be the task performance of model f on clean and adversarially perturbed images, respectively. With multiple adversarial methods $\{adv_i\}_i$ and corresponding task performances $\{perf_{adv_i}(f)\}_i$, we can obtain an overall performance metric as the mean or worst-case performance.

Robust task performance: Model performance $perf_{adv}(f)$ on attacked images. Multiple models are ranked by $perf_{adv}(f)$ for comparison.

To evaluate model robustness with respect to a baseline, [61] proposes effectiveness and relative robustness metrics. These metrics are originally proposed for natural distribution shifts in image classification, but can be readily adapted to our task with our definitions of $perf_{clean}(f)$ and $perf_{adv}(f)$.

Effective robustness: Let $\beta(perf_{clean}(f))$ be the baseline performance on clean images. [61] instantiates β as a log-linear function on $perf_{clean}(f)$ across models $\{f_m\}_m$. Then the effective robustness of a model f is:

$$\rho_{eff}(f) = perf_{adv}(f) - \beta(perf_{clean}(f)). \tag{8}$$

Models with special robustness properties will have $perf_{adv}(f))$ above the β fit.

Relative robustness: To compare f_1 with f_2, relative robustness is computed as:

$$\rho_{rel}(f) = perf_{adv}(f_1) - perf_{adv}(f_2). \tag{9}$$

f_1 and f_2 can be two different models (for example, one of them can be a baseline model), or two versions of a model with and without robustness intervention.

G. Datasets and Models

We summarize the datasets and computer vision models used to evaluate adversarial attacks for object detection in existing literature in Tables V and VI, respectively. The common datasets used are PASCAL VOC [62] and MS COCO [34]. The common models tested are anchor-based detectors such as SSD [37], YOLO v2-v4 [38], [63], [64] and Faster R-CNN [39].

TABLE V: Models used in literature to evaluate adversarial attacks in object detection.

Detector Type	Model / Algorithm	Version / Backbone
One-stage Anchor-based	SSD	SSD300 VGG-16, SSD512 VGG-16
	YOLOv2	Darknet-19
	YOLOv3	Darknet-53
	YOLOv4	CSPDarknet-53
	RetinaNet	ResNet-101
	EfficientDet	EfficientNet + BiFPN
Two-stage Anchor-based	R-FCN	ResNet-101
	Faster R-CNN	ZFNet, VGG-16, ResNet-50, ResNet-101, ResNeXt-101
	Libra R-CNN	ResNet-50
	Mask R-CNN	ResNeXt-101
	Cascade R-CNN	ResNet-101
	Cascade Mask R-CNN	ResNeXt-101
	Hybrid Task Cascade	ResNeXt-101
One-stage Anchor-free	FCOS	ResNet-50
	FoveaBox	ResNet-50
	FreeAnchor	ResNet-50
	DETR	DETR ResNet-50, Deformable DETR ResNet-50

TABLE VI: Datasets used in literature to evaluate adversarial attacks for object detection. † denotes that dataset requires additional annotation.

Data Type	Classes	Dataset
Image	Common objects	VOC-2007, VOC-2012, COCO, KITTI
	Person	Inria Person, CityPersons, ImageNet†
	Traffic sign	Stop Sign, Mapilliary†
Video	Common objects	ImageNet VID

IV. Defenses against Adversarial Attacks

Extensive research efforts have investigated defense mechanisms against adversarial attacks in image classification, and we refer readers to existing survey papers [65], [66] for a comprehensive overview. The defensive strategies include adversarial training, knowledge distillation, gradient regularization, network architecture search for robust model designs, auxiliary detector networks for adversarial example identification, denoising adversarial inputs and feature representations, and adversarial purification through generative models to mitigate the impact of perturbations.

In comparison, literature on defenses in object detection is more limited. Dong *et al.* [13] reviewed the defenses focused on applications in autonomous vehicles. In general, defense strategies can be categorized into three primary groups: 1) training procedure modifications, 2) input/feature denoising, and 3) external attack detection networks. In the first group, Dong *et al.* [67] improved the training procedure to disentangle model gradients on clean and adversarial samples during training, while Saha *et al.* [68] developed methods to limit the dependence on spatial context during training. Chen *et al.* [69] introduced class-wise loss normalization to balance the influence of each object class, and Zhang and Wang [70] proposed to generate attack samples using multiple loss components for adversarial training. Chiang *et al.* [71] developed the method to certify object detection predictions by aggregating multiple predictions for randomly perturbed inputs via median smoothing. For the input denoising approach, Zhou *et al.* [72] developed the method to transform the input images to remove non-robust features and adversarial noise. To defend against patch attacks, Liu *et al.* [73] introduced the method to localize adversarial patches and remove them from the images, while Kim *et al.* [74] presented the approach to eliminate adversarial features through learned detection mechanisms. The third group comprises detection-based defenses that aim to identify adversarial samples and trigger appropriate alerts. For instance, Strack *et al.* [75] and *et al.* [76] focused on training samples with randomly added patches to detect adversarial patches. Xiang and Mittal *et al.* [77] ultilized an image classifier and object detector to validate prediction objectness, where predictions lacking sufficient explanatory support are flagged as potential attacks.

V. Evaluation

This section thoroughly evaluates the open-sourced adversarial attacks listed in Table III. Our experiments aim to compare available open-source attack methods and provide a comprehensive analysis of their performances across various object detector architectures.

Specifically, section V-B1 evaluates and compares the attack effectiveness of 5 open-source attack methods, including Dpatch [54], TOG [18], DPAttack [41], CAT [33], EBAD [20]. For TOG [18], we reimplemented the algorithms from TensorFlow to PyTorch to evaluate them with MMDetection's pretrained object detectors, in a manner consistent with other attacks in our experiment. For DPatch [54], we used the reimplementation from Adversarial Robustness Toolbox (ART), an open-source Python library for Machine Learning Security [1]. Other methods were excluded due to outdated or unmaintained code, or the absence of a well-specified environment and setup for rerunning. We evaluate these attacks on Faster R-CNN and YOLOv3, two object detectors that are commonly used in literature. Section V-B2 focuses on studying cross-model black-box attacks on 7 different object detectors with 49 surrogate-victim pairs. We highlight the effects of different detector architectures on the attack performance and their robustness against the attacks. Lastly, section V-B3 provides a

[1] https://github.com/Trusted-AI/adversarial-robustness-toolbox

transferability study of ensemble-based attacks on 26 object detectors, ranging from traditional to modern state-of-the-art ones, including detectors with vision-language pretraining which have not been thoroughly studied in existing literature.

A. Experiment Setup

For a fair comparison, we standardized attack hyperparameters including the maximum number of queries and perturbation budget, and ran all attacks on a single GeForce GTX 1080 12GB RAM.

Dataset: We conduct evaluations in sections V-B1 and V-B3 using the COCO 2017 validation set with 5000 images on 80 object classes. For the evaluation on ensemble attacks in section V-B2, we use a subset of 500 random images (10%) from the COCO 2017 validation set to evaluate 49 surrogate-victim pairs. Using a smaller subset helps to reduce the time and cost of running this evaluation.

Object detectors: We evaluate attacks on object detectors commonly used in literature: Faster-RCNN with a ResNet-50 backbone (denot. FR-RN50) and YOLOv3 with a Darknet-53 backbone (denot. YOLOv3-D53). We also include other object detectors, such as RetinaNet, Libra R-CNN and FCOS, and modern state-of-the-art detectors such as DeTR with Transformer architecture, Grounding DINO, and GLIP with vision-language pretraining.

We point out that different frameworks (Torch, MMDetection, Tensorflow) may provide different model pretrained weights, and these weights may also be periodically updated. Thus, even using the same detector architecture may yield mAP scores different from those originally reported in the enlisted papers in Table III. For a fair comparison, we use the pretrained model weights provided by MMDetection v3.3.0 pretrained on COCO 2017 train set for all evaluations. Their mAP on clean images are presented in Table VII.

TABLE VII: The mAP score of FR-RN50 and YOLOv3 evaluated on the clean COCO 2017 validation set.

Model	Dataset	mAP Clean (%)
FR-RN50	COCO 2017	58.10
YOLOv3-D53	COCO 2017	52.80

Maximum queries and perturbation budget: We refer to *query* as the number of times the attack algorithm retrieves the detection result from the victim detector to update its perturbation. Note that this is typically called an *iteration* in white-box settings. By default, we limit the maximum number of queries to 10. Several works allowed as many as 5,000 - 100,000 queries, which is impractical as repeated queries may be easily detected by defense mechanisms. We also reported attack performances under a single query run. We set $L_\infty = 10/255$, a typical value set in literature.

Metrics: We calculate the mAP of the attacked images and compare it to the mAP of the clean images. Given that different versions of pretrained object detectors may exhibit different mAP scores on clean images, we provide the *percentage drop in mAP* as a normalized metric for easy comparison, calculated by $(1 - \frac{mAP\ Adv.}{mAP\ Clean}) * 100\%$. We also report the *time cost*, defined as the time (in seconds) needed to generate the perturbations, including the iterative updating process.

Note that not all adversarial attacks aim to reduce the mAP score, thus their observed mAP drops might not appear substantial. Also, while *fooling rate* is a popular metric, we do not include it in our evaluation. This is because each work defines the criteria for successful attacks differently based on their specific setup and objectives, making the fooling rate non-comparable across different attacks.

B. Results and Analysis

1) Attack effectiveness: Tables VIII and IX show the attack performances on victims YOLOv3-D53 and FR-RN50 respectively.

TOG is an effective white-box attack with low time cost: White-box setting assumes access to all detection information, such as prediction scores, losses, and bounding box coordinates. Thus, the resulting mAP drops in this setup are often more significant compared to gray-box and black-box attacks. Our experiment finds that TOG-untargeted is an effective white-box attack that achieves mAP drops of more than 50% with only 10 queries. In our evaluation, the TOG untargeted attack results in the highest mAP drops of 55.59% on FR-RN50 and 60.22% on YOLOv3-D53. It also has the lowest time cost compared to other attacks, with less than 4 seconds to attack FR-RN50 and less than 1 second to attack YOLOv3-D53. Older methods such as DAG [31] or UEA [49] require more than 30 queries yet still achieve poorer results [78].

For patch attacks under white-box setting, DPatch is shown to be ineffective under our experiment setting with an insignificant mAP drop of only 6.37 %. DPatch requires a large number of queries (iterations), between 1000 and 20000 [54], which explains why it falls short in our experiment with a limited but more realistic 10 queries. Its attack success is also sensitive to the location of the patch. If the patch is placed where objects are present in the image, the chance of success increases. On the other hand, DPattack produces a significant mAP drop of 22.68 %. Nevertheless, a common issue with patch attacks is that the perturbed patch added to the image is easily recognizable by human eyes, making them easy to be detected in real-world scenarios.

EBAD is specifically designed for ensemble-based black-box attacks, yet still shows a significant mAP drop in the white-box setting with a single surrogate. In Tables VIII and IX, EBAD-single refers to using the same surrogate model (FR-RN50 or YOLOv3-D53) to attack the same victim architecture, achieving mAP drops of 22.37% on FR-RN50 and 21.97% on YOLOv3-D53.

For the ensemble-based CAT and EBAD, we use a group of two surrogates FR-RN50 and YOLOv3, and separately test the results on 2 victim detectors FR-RN50 and YOLOv3-D53. CAT, a transfer attack method that generates a universal perturbation from surrogates to attack unseen victim detectors, achieves approximately 16% mAP drops. Our experiment shows that EBAD-ensemble achieves impressive mAP drops of more than 22.55% within 10 queries. Moreover, its attack

TABLE VIII: Results of adversarial attack on FR-RN50. L_∞ denotes whether the attack has constrained $L_\infty = 10/255$. For the patch attack DPatch, the pixel change can vary from 0.0 to 255.0. Attacks with *ensemble* indicates that these methods use a group of two surrogates FR-RN50 and YOLOv3-D53. EBAD single is the EBAD attack with a single surrogate FR-RN50. The sign * denotes reimplemented methods.

FR-RN50 victim									
Attack name	L_∞	**10-query**				**Single query**			
		mAP Drop (%)	mAP Adv. (%)	Time (s)	PSNR	mAP Drop (%)	mAP Adv. (%)	Time (s)	PSNR
DPatch *	✗	6.37	54.40	4.51	36.07	4.82	55.30	0.21	36.21
TOG untargeted *	✓	55.59	25.80	3.86	32.13	16.87	48.30	0.80	32.79
TOG fabricated *	✓	14.77	45.00	8.66	32.02	11.53	51.40	1.12	32.76
TOG vanishing *	✓	30.80	40.20	6.94	32.20	22.89	44.80	0.76	32.78
DPAttack	✗	19.96	46.50	2.37	25.57	19.44	46.80	0.33	25.62
CAT ensemble	✓	16.69	48.40	28.55	33.26	6.37	54.40	4.18	33.31
EBAD single	✓	22.37	45.10	20.03	31.83	24.26	44.00	5.10	32.05
EBAD ensemble	✓	22.55	45.00	32.32	31.76	23.58	44.40	3.75	31.96

TABLE IX: Results of adversarial attack on FR-RN50. L_∞ denotes whether the attack has constrained $L_\infty = 10/255$. For the patch attack DPatch, the pixel change can vary from 0.0 to 255.0. Attacks with *ensemble* indicates that these methods use a group of two surrogates FR-RN50 and YOLOv3-D53. EBAD single is the EBAD attack with a single surrogate YOLOv3-D53. The sign * denotes reimplemented methods.

YOLOv3-D53 victim									
Article	L_∞	**10-query**				**Single query**			
		mAP Drop (%)	mAP Adv. (%)	Time (s)	PSNR	mAP Drop (%)	mAP Adv. (%)	Time (s)	PSNR
DPatch *	✗	3.97	50.70	1.04	35.18	2.65	51.40	0.12	36.04
TOG untargeted *	✓	60.22	21.70	0.85	32.14	14.77	45.00	0.31	32.71
TOG fabricated *	✓	27.46	38.30	0.85	32.19	13.63	45.60	0.31	32.71
TOG vanishing *	✓	39.20	32.10	1.16	32.23	24.43	39.90	0.16	32.76
DPAttack	✗	22.68	45.00	2.37	25.57	22.68	45.00	0.33	25.62
CAT ensemble	✓	16.09	44.30	28.55	33.26	8.90	48.10	4.18	33.31
EBAD single	✓	21.97	41.20	7.65	31.79	25.75	39.20	1.47	32.07
EBAD ensemble	✓	19.70	42.40	27.27	31.71	19.69	42.40	3.36	32.01

performance with a single query also outperforms all other methods. We also notice that the mAP drops from EBAD's single query attacks are slightly higher than those from the 10-query attacks. This could potentially be due to their approach of adapting the PGD [79] optimization from white-box adversarial attacks for image classification, which aims to reduce accuracy scores rather than directly targeting mAP scores. From what we observed, within the first iterations of PGD, the average perturbation values were typically at the maximum $L_\infty = 10$ constraint. However, after several iterations, they were decreased to around 5, making the perturbation less visible to the human eye while still focusing on accuracy reduction. This explains why for the EBAD, increasing the number of queries may not necessarily lead to higher mAP drops in our experiment.

Earlier black-box attacks such as PRFA [23] and RAD [19] require several constraints to be successful. They both require knowing the detector architecture to adjust its optimization process. Moreover, RAD sets a larger perturbation norm $L_\infty = 16/255$. Meanwhile, ensemble-based methods such as CAT [33] and EBAD [20] use a group of surrogate detectors to attack another victim's black-box detector that requires no prior knowledge about the victim, which is a more practical use case.

2) Cross-model black-box attack: In this experiment, we study the attack effictiveness of different surrogate-victim detector pairs in the black-box setting. Table X evaluates cross-model black-box attack performance with different sets of attack and victim models ranging from traditional detectors (Faster R-CNN, YOLOv3, RetinaNet, Libra R-CNN, FCOS) to modern state-of-the-art detectors such as DeTR with Transformer architecture, and GLIP with vision-language pretraining. We conduct our experiment with EBAD attack due to its potential wider applicability as an imperceptible black-box attack. We use a subset of 500 samples from the COCO 2017 validation set for evaluation.

Robustness against white-box and robustness against black-box attacks are not correlated. We observe that stronger detectors, such as DeTR and GLIP, which achieve higher mAP on clean images also tend to be relatively robust against adversarial attacks. In particular, DeTR-RN50 and GLIP Swin-T have respective white-box mAP drops of 8.01% and 14.85%, the lowest amongst the models tested in Table X. Interestingly however, models that are vulnerable against white-box attacks can still be robust against black-box attacks. YOLOv3-MN has the highest white-box mAP drop of 43.38% but the lowest black-box mAP drop of 4.41%. DeTR-RN50 attacked by YOLOv3-MN has a mAP drop of 8.17%, while YOLOv3-MN attacked by DeTR-RN50 has a

TABLE X: The mAP score drop (%) of EBAD attack with different surrogate and victim detectors. Entries on the diagonal are equivalent to white-box attacks as they use the same detectors for the attacker and victim. The *Mean mAP drop* column reports the mean mAP drop for each victim with different surrogates.

Victim \ Surrogate	FR -RN50	YOLOv3 -D53	YOLOv3 -MN	RetinaNet -RN50	FCOS -RN50	DeTR -RN50	GLIP Swin-T	Mean mAP drop	Mean black-box mAP drop
FR-RN50	**17.44**	10.19	7.77	13.99	10.02	8.98	8.81	11.03	9.96
YOLOv3-D53	6.03	**22.83**	6.22	3.77	4.72	4.90	4.90	7.63	5.10
YOLOv3-MN	3.79	7.58	**43.38**	2.95	4.63	3.37	4.21	9.98	4.41
RetinaNet-RN50	19.05	10.34	7.10	**27.40**	10.52	11.25	8.35	13.43	11.10
FCOS-RN50	6.57	6.07	6.56	7.72	**18.27**	6.90	5.91	8.28	6.62
DeTR-RN50	7.37	8.01	8.17	8.81	7.21	**8.01**	7.53	7.87	7.85
GLIP Swin-T	4.90	5.17	4.90	4.22	5.31	3.00	**14.85**	6.05	4.58

lower mAP drop of 3.37%.

Using a surrogate with the same backbone architecture with the victim does not necessarily result in a high mAP drop. In Table X, 5 models use a ResNet-50 backbone. We observe that attacking a victim model with a surrogate that shares the backbone architecture generally results in comparatively higher mAP drop, but the value of mAP drop is not consistently high. For instance, RetinaNet-RN50 have mAP drops above 10% (and up to 19.05%) when attacked by surrogates with ResNet-50 backbone, while FCOS-RN50 and DeTR-RN50 have no mAP drops above 10%.

3) Transferability study: In this experiment, we evaluate the transferability of the ensemble-based attack EBAD on different unseen detectors. Among those, Grounding DINO [80] and GLIP [81] are state-of-the-art detectors that achieve remarkably high detection results on various benchmark datasets, including COCO 2017.

We conduct this experiment on the full COCO 2017 validation set, which includes 5000 samples. We assess cross-model transferability using 4 sets of perturbed images, generated from different surrogate groups as follows:

- Images perturbed by 2 surrogates FR-RN50 and YOLOv3-D53, using RetinaNet-RN50 as the victim.
- Images perturbed by 4 surrogates: one-stage anchor-based models (YOLOv3-D53, SSD), a two-stage anchor-based model (FR-RN50), and a one-stage anchor-free model (FCOS), with RetinaNet-RN50 as the victim.
- Images perturbed by 2 surrogates YOLOv3-D53 and GLIP Swin-T, with RetinaNet-RN50 as the victim.
- Images with random noise of $L_\infty = 10$, provided as a baseline for comparison.

The output images from these sets of attacks on RetinaNet-RN50 are then tested on 26 unseen detectors. In Table XI, we report the % mAP drops of these unseen detectors for different object areas, following the criteria from MMDetection:

- Small objects: Areas in an image are less than 1024 (32x32 pxiels)
- Medium objects: Areas in an image are from 1024 to 9216 (32x32 to 96x96 pixels)
- Large objects: Areas in an image greater than 9216 (96x96 pixels)

All of the object detectors in this evaluation are pretrained on COCO 2017.

Detections on smaller objects are more vulnerable to adversarial attacks. The mAP drops reported for small objects are significantly higher compared to medium or large objects when under attack. More traditional victim detectors, such as Faster R-CNN, YOLOv3, RetinaNet, Libra R-CNN, and FCOS, experience mAP drops for small objects of 10%-20% greater than those of the larger-sized objects. For newer victim detectors such as DeTR, RTMDet, GLIP, and Grounding DINO, the gaps are smaller yet remain significant.

Transferred attacks show decreased effects on more recent detector architectures. There is a clear trend that the mAP scores decline less for more recent object detectors compared to older ones. More modern object detector may be more robust as they use novel architectures and are trained on larger data. RTMDet, GLIP, and Grounding DINO experience considerably smaller mAP drops, all below 10%. Even when using GLIP Swin-T as the surrogate, the victims including GLIP itself, RTMDet, and Grounding DINO generally suffer smaller mAP drops.

A diverse group of surrogate detectors increases the adversarial attack transferability. Utilizing an ensemble of 4 surrogate detectors results in a greater mAP drop across all unseen detectors. A broader range of surrogate models with diverse architectures and types can significantly increase the effectiveness of adversarial attacks. From Table XI, we also show that the 4-surrogate group results in greater mAP drops than YOLOv3-D53 + GLIP Swin-T on new victim detectors such as RTMDet, GLIP, and Grounding DINO.

VI. LIMITATIONS

While we made every effort to include all relevant adversarial attacks in our experiments, some methods were excluded due to outdated or unmaintained code, or the absence of a well-specified environment and setup for rerunning. For TOG [18], we reimplemented the algorithms from TensorFlow to PyTorch to evaluate them with MMDetection's pretrained object detectors, in a manner consistent with other attacks in our experiment. We reproduced the algorithm as best as we can based on the descriptions in the original manuscript [18]. For another work, DPatch [54], we used the reimplementation from Adversarial Robustness Toolbox (ART), an open-source Python library for Machine Learning Security. Consequently, the results presented may differ from those reported in the original papers. Other methods were not included due to insufficient documentation on parameter setups. Despite attempts to

TABLE XI: The mAP score drop (%) of EBAD attack on unseen victim detectors. *Transfer* distinguishes evaluation on unseen victims (✓) with seen victims (✗). *All* reports the normal % mAP drops at IoU 0.5 for all-sized objects. *Small, Medium, Large* reports the % mAP drops at IoU 0.5:0.95 on small, medium, and large objects respectively. *Noise* reports the % mAP drop from random noise.

Victim model	Backbone	Noise	FR-RN50 + YOLOv3-D53					FR-RN50 + YOLOv3-D53 + FCOS-RN50 + SSD					YOLOv3-D53 + GLIP Swin-T				
		All	Transfer	All	Small	Medium	Large	Transfer	All	Small	Medium	Large	Transfer	All	Small	Medium	Large
Faster R-CNN	Resnet-50	5.85	✗	27.71	37.26	27.32	27.65	✗	35.63	41.03	34.87	33.13	✓	13.77	24.53	15.37	11.23
	Resnet-101	5.32	✓	15.14	27.23	16.93	13.50	✓	23.29	33.04	25.86	20.94	✓	12.15	27.23	13.73	9.39
	ResneXt-101	5.31	✓	14.49	25.00	15.38	12.71	✓	22.70	31.25	24.83	21.49	✓	12.08	24.17	14.07	10.28
YOLOv3	Darknet-53	2.84	✗	21.59	20.83	17.66	22.37	✗	25.38	23.61	22.15	28.41	✗	16.10	16.67	12.87	15.66
	MobileNet-V2	2.41	✓	7.03	15.09	6.77	6.57	✓	9.45	15.10	8.76	8.28	✓	5.93	4.72	5.58	4.86
RetinaNet	Resnet-50	5.78	✗	23.29	33.82	23.57	23.49	✗	31.95	38.23	32.75	33.89	✓	14.62	25.98	16.13	13.51
	Resnet-101	5.20	✓	15.45	28.00	16.35	12.87	✓	23.09	35.02	24.77	21.58	✓	12.50	26.27	13.55	10.50
Libra R-CNN	Resnet-50	5.21	✓	19.83	27.60	20.95	19.17	✓	27.56	33.48	29.28	26.60	✓	12.10	22.17	13.81	10.31
FCOS	Resnet-50	4.41	✓	11.76	20.81	13.94	11.11	✗	26.80	30.61	27.89	30.11	✓	9.80	20.00	12.42	9.32
	Resnet-101	4.79	✓	14.70	25.78	16.05	14.17	✓	24.27	33.33	26.05	24.75	✓	11.45	22.67	13.49	11.58
	ResneXt-101	4.96	✓	13.76	26.15	15.05	12.43	✓	20.32	29.23	23.01	20.29	✓	11.20	24.23	13.98	10.05
DeTR	Resnet-50	5.54	✓	16.61	27.98	17.61	17.52	✓	25.47	33.20	27.25	26.87	✓	12.50	24.63	13.42	11.90
RTMDet-T	-	3.80	✓	8.46	12.38	9.67	8.23	✓	12.26	16.67	13.63	12.86	✓	7.25	14.29	8.79	6.86
RTMDet-M	-	4.05	✓	9.75	19.21	9.61	7.07	✓	11.39	22.47	12.94	10.22	✓	7.65	18.24	9.06	6.02
RTMDet-L	-	3.78	✓	7.41	17.06	9.25	5.40	✓	9.88	23.25	11.57	8.32	✓	6.98	18.53	8.72	4.96
RTMDet-X	-	3.83	✓	7.53	18.89	8.54	5.64	✓	9.80	19.17	11.15	7.22	✓	7.10	17.78	8.54	4.77
GLIP	swin-T (A)	4.06	✓	9.10	18.16	10.71	9.02	✓	13.57	21.48	15.54	14.44	✗	12.17	20.72	14.16	11.73
	swin-T (B)	3.46	✓	7.33	16.20	8.38	6.79	✓	11.34	19.49	13.50	11.96	✗	9.96	20.51	11.97	9.45
	swin-T (C)	3.40	✓	6.80	15.16	8.18	6.37	✓	10.20	18.83	12.35	10.13	✗	10.20	19.80	12.85	9.99
	swin-T	3.26	✓	7.07	14.10	8.67	6.29	✓	10.47	20.30	13.17	10.30	✗	12.93	22.03	15.33	13.73
	swin-L	1.94	✓	4.26	12.95	5.32	3.23	✓	6.33	16.07	8.92	4.31	✓	4.78	13.62	6.89	2.96
GDINO	swin-T	2.91	✓	5.82	13.88	7.28	4.27	✓	9.14	16.94	11.16	7.99	✓	7.95	16.94	9.55	6.20
	swin-B	2.45	✓	4.12	11.89	5.49	1.46	✓	6.06	14.19	8.00	3.46	✓	4.90	13.27	6.44	1.86

contact the authors, we were unable to obtain the necessary information and code base to execute the methods.

As mentioned in Section V-A, we standardized the number of queries and perturbation budget (L_∞) to ensure comparability among different methods. As a result, methods such as DPatch [54] performed less effectively, as they were originally designed for larger query settings. The original implmentation of DPatch used up to 200k iterations, which is impractical especially for real time attacks. Moreover, variations in L_∞ values can significantly impact the mAP outcome. Higher L_∞ values generally result in a more substantial mAP drop, but they also make the perturbations more detectable to the human eye. Certain patch-based methods, such as DPatch and DPAttack, were easily visible to human observers and did not adhere to this L_∞ constraint by design. Imposing a constrained L_∞ on these methods nullified their attack effectiveness. Therefore, we did not set the constraint for these two methods in our experiment.

VII. Open areas for research

Adversarial attacks on modern object detectors have more room for research. In our experiments, we observed that more recent object detectors, such as Transformer-based and vision-language pretrained ones, are potentially more robust against adversarial attacks. Modern object detectors may be more robust as they use novel architectures, training strategies and are trained on different or larger datasets. Prior research [82]–[84] focused on image recognition found that the robustness of transformers can be partially attributed to their training strategies and self-attention mechanisms. More research is needed on the architectural, training and dataset designs that significantly affect robustness for object detection.

Defences on small objects are particularly needed. We observed that small objects within images are significantly more vulnerable to adversarial attacks. To our knowledge, there is currently no research specifically addressing adversarial attacks on small objects, either from the attack or defense perspective. We recommend further investigation into this issue in the future.

Adversarial attacks based on image rendering are underexplored for the object detection task. Besides pixel value perturbations covered in Section II, literature in image classification also explored attacks based on image rendering [85]–[87], including image rotation, translation, lighting, and coloring. That is, an adversarial attack is to render image x such that the transformed image $x' = r(x)$ for a rendering function $r(\cdot)$ is incorrect i.e. $f(x'; \Theta) = f(r(x); \Theta) \neq y$. Rendering-based attacks have practical implications in applications such as autonomous vehicles and surveillance systems, where adversarial rendering methods can simulate adverse weather conditions, camera angles and object orientations to test model robustness in constantly shifting environments.

Development in multimodal models presents new areas for research on adversarial attacks and robustness. For instance, applications such as autonomous driving and robotics require additional temporal and depth information. In these applications, it is common for models to integrate RGB images with other data sources such as LiDAR and 3D point clouds, and to process video data instead of a single image. These multimodal inputs offer richer contextual information, but also introduce new attack surfaces. For instance, adversaries could manipulate depth information from LiDAR or distort 3D point cloud data to mislead object detection systems, even when the RGB input appears unaffected. Similarly, perturbing temporal consistency in video data could degrade detection models that

rely on motion cues. Research in this area is crucial to develop robust defense mechanisms capable of handling coordinated multimodal attacks.

Physical adversarial attacks in object detection need benchmark evaluation. Conducting evaluations in the physical world is often expensive and logistically challenging, particularly when it comes to automatically generating attacks for testing under various real-world conditions. The physical environment imposes constraints due to lighting conditions, occlusions, camera angles, and object distances, all of which can significantly affect the effectiveness of adversarial attacks and model robustness. While augmentations such as Expectation Over Transformation (EOT) have been employed to simulate real-world transformations, ensuring the fidelity of these simulated images to real-world scenarios remains a crucial challenge for accurate and reliable evaluation.

VIII. Conclusion

This paper provides a comprehensive survey and evaluation of adversarial attacks in object detection, revealing critical insights into the existing attack methods. While adversarial attacks in image classification are extensively studied, their impact on object detection presents distinct challenges that require dedicated investigation. Hence, in this paper, we reviewed different types of existing adversarial attacks on object detection and common evaluation metrics and conducted comprehensive experiments on evaluating the effectiveness and transferability of the attacks on different object detectors. Our experiments revealed several key findings: detector robustness against white-box and black-box attacks shows no correlation, and using surrogate models with matching backbones does not guarantee better transferability. Smaller objects proved more vulnerable to adversarial attacks, while newer detector architectures demonstrated improved resistance. We also found that diverse surrogate detector ensembles significantly increase attack transferability. Moving forward, key priorities include developing standardized evaluation protocols, conducting comprehensive transferability studies, and investigating the relationships between model architecture and adversarial vulnerability, particularly concerning physical-world attacks. As object detection systems increasingly power critical applications, such understanding becomes essential for building robust and reliable systems.

References

[1] S. Srivastava and G. Sharma, "Omnivec: Learning robust representations with cross modal sharing," in *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2024.

[2] P. Wang, S. Wang, J. Lin, S. Bai, X. Zhou, J. Zhou, X. Wang, and C. Zhou, "ONE-PEACE: Exploring one general representation model toward unlimited modalities," *CoRR*, 2023. [Online]. Available: https://arxiv.org/abs/2305.11172

[3] Z. Geng, C. Wang, Y. Wei, Z. Liu, H. Li, and H. Hu, "Human pose as compositional tokens," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[4] J. Ding and Z. Xu, "Adversarial attacks on deep learning models of computer vision: A survey," in *International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP)*, 2020.

[5] X. Huang, D. Kroening, W. Ruan, J. Sharp, Y. Sun, E. Thamo, M. Wu, and X. Yi, "A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability," *Computer Science Review*, vol. 37, p. 100270, 2020.

[6] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14 410–14 430, 2018.

[7] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay, "A survey on adversarial attacks and defences," *CAAI Transactions on Intelligence Technology*, vol. 6, 03 2021.

[8] W. Tan, J. Zhao, X. Liang, H. Lu, B. Song, and H. Guan, "Adversarial example attack and defence of object recognition: A survey," in *IEEE International Conference on Unmanned Systems (ICUS)*, 2022.

[9] C. Wang, M. Zhang, J. Zhao, and X. Kuang, "Black-box adversarial attacks on deep neural networks: A survey," in *International Conference on Data Intelligence and Security (ICDIS)*, 2022.

[10] F. Croce, M. Andriushchenko, V. Sehwag, E. Debenedetti, N. Flammarion, M. Chiang, P. Mittal, and M. Hein, "RobustBench: a standardized adversarial robustness benchmark," in *Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2021.

[11] S. Tang, R. Gong, Y. Wang, A. Liu, J. Wang, X. Chen, F. Yu, X. Liu, D. Song, A. Yuille, P. H. Torr, and D. Tao, "RobustART: Benchmarking robustness on architecture design and training techniques," *CoRR*, 2021. [Online]. Available: https://arxiv.org/abs/2109.05211

[12] F. Ren, Y. Yang, C. Hu, Y. Zhou, and S. Ma, "ADVRET: An adversarial robustness evaluating and testing platform for deep learning models," in *2021 IEEE 21st International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, 2021, pp. 9–14.

[13] A. Amirkhani, M. Karimi, and A. Banitalebi-Dehkordi, "A survey on adversarial attacks and defenses for object detection and their applications in autonomous vehicles," *The Visual Computer*, vol. 39, 2022.

[14] J.-X. Mi, X.-D. Wang, L.-F. Zhou, and K. Cheng, "Adversarial examples based on object detection tasks: A survey," *Neurocomputing*, vol. 519, no. C, p. 114–126, 2023.

[15] B. Xu, J. Zhu, and D. Wang, "Adversarial attacks for object detection," in *Chinese Control Conference (CCC)*, 2020, pp. 7281–7287.

[16] N. Hingun, C. Sitawarin, J. Li, and D. Wagner, "REAP: A large-scale realistic adversarial patch benchmark," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[17] Z. Wu, S.-N. Lim, L. S. Davis, and T. Goldstein, "Making an invisibility cloak: Real world adversarial attacks on object detectors," in *European Conference on Computer Vision (ECCV)*, 2020.

[18] K.-H. Chow, L. Liu, M. Loper, J. Bae, M. E. Gursoy, S. Truex, W. Wei, and Y. Wu, "Adversarial objectness gradient attacks in real-time object detection systems," in *IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, 2020.

[19] S. Chen, F. He, X. Huang, and K. Zhang, "Relevance attack on detectors," *Pattern Recognition*, vol. 124, p. 108491, 2022.

[20] Z. Cai, Y. Tan, and M. S. Asif, "Ensemble-based blackbox attacks on dense prediction," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[21] Z. Cai, S. Rane, A. E. Brito, C. Song, S. V. Krishnamurthy, A. K. Roy-Chowdhury, and M. S. Asif, "Zero-query transfer attacks on context-aware object detectors," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[22] J. Wang, C. Cui, X. Wen, and J. Shi, "TransPatch: A transformer-based generator for accelerating transferable patch generation in adversarial attacks against object detection models," in *Computer Vision – ECCV 2022 Workshops*, 2022.

[23] S. Liang, B. Wu, Y. Fan, X. Wei, and X. Cao, "Parallel rectangle flip attack: A query-based black-box attack against object detection," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[24] D. Wang, C. Li, S. Wen, Q.-L. Han, S. Nepal, X. Zhang, and Y. Xiang, "Daedalus: Breaking nonmaximum suppression in object detection via adversarial examples," *IEEE Transactions on Cybernetics*, vol. 52, no. 8, pp. 7427–7440, 2022.

[25] T. Du, S. Ji, B. Wang, S. He, J. Li, B. Li, T. Wei, Y. Jia, R. Beyah, and T. Wang, "DetectSec: Evaluating the robustness of object detection models to adversarial attacks," *International Journal of Intelligent Systems*, vol. 37, no. 9, pp. 6463–6492, Sep. 2022.

[26] D. Y. Yang, J. Xiong, X. Li, X. Yan, J. Raiti, Y. Wang, H. Wu, and Z. Zhong, "Building towards "invisible cloak": Robust physical adversarial attack on YOLO object detector," in *IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, 2018, pp. 368–374.

[27] E. Arani, S. Gowda, R. Mukherjee, O. Magdy, S. S. Kathiresan, and B. Zonooz, "A comprehensive study of real-time object detection networks across multiple domains: A survey," *Transactions on Machine Learning Research*, 2022.

[28] K.-H. Chow, L. Liu, M. E. Gursoy, S. Truex, W. Wei, and Y. Wu, "Understanding object detection through an adversarial lens," in *European Symposium on Research in Computer Security*, 2020.

[29] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," in *International Conference on Machine Learning (ICML)*, 2017.

[30] S. Thys, W. V. Ranst, and T. Goedeme, "Fooling automated surveillance cameras: Adversarial patches to attack person detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE Computer Society, 2019.

[31] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille, "Adversarial examples for semantic segmentation and object detection," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

[32] M. Yin, S. Li, C. Song, M. S. Asif, A. K. Roy-Chowdhury, and S. V. Krishnamurthy, "ADC: Adversarial attacks against object detection that evade context consistency checks," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2022.

[33] Z. Cai, X. Xie, S. Li, M. Yin, C. Song, S. V. Krishnamurthy, A. K. Roy-Chowdhury, and M. S. Asif, "Context-aware transfer attacks for object detection," in *AAAI Conference on Artificial Intelligence*, 2022.

[34] T.-Y. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *European Conference on Computer Vision (ECCV)*, 2014.

[35] A. Shapira, A. Zolfi, L. Demetrio, B. Biggio, and A. Shabtai, "Phantom sponges: Exploiting non-maximum suppression to attack deep object detectors," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2023.

[36] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[37] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *European Conference on Computer Vision (ECCV)*, 2015.

[38] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *CoRR*, 2020. [Online]. Available: https://arxiv.org/abs/2004.10934

[39] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 1137–1149, 2015.

[40] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra R-CNN: Towards balanced learning for object detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[41] S. Wu, T. Dai, and S. Xia, "DPAttack: Diffused patch attacks against universal object detection," *CoRR*, 2020. [Online]. Available: https://arxiv.org/abs/2010.11679

[42] T. Kong, F. Sun, H. Liu, Y. Jiang, L. Li, and J. Shi, "FoveaBox: Beyond anchor-based object detector," *IEEE Transactions on Image Processing*, pp. 7389–7398, 2020.

[43] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: A simple and strong anchor-free object detector," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 4, pp. 1922–1933, 2022.

[44] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European Conference on Computer Vision (ECCV)*, 2020.

[45] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," in *International Conference on Learning Representations (ICLR)*, 2021.

[46] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[47] K. Duan, L. Xie, H. Qi, S. Bai, Q. Huang, and Q. Tian, "Corner proposal network for anchor-free, two-stage object detection," *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[48] P. Sun, R. Zhang, Y. Jiang, T. Kong, C. Xu, W. Zhan, M. Tomizuka, L. Li, Z. Yuan, C. Wang *et al.*, "Sparse r-cnn: End-to-end object detection with learnable proposals," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 14 454–14 463.

[49] X. Wei, S. Liang, N. Chen, and X. Cao, "Transferable adversarial attacks for image and video object detection," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2019.

[50] H. Huang, Y. Wang, Z. Chen, Z. Tang, W. Zhang, and K.-K. Ma, "RPAttack: Refined patch attack on general object detectors," in *IEEE International Conference on Multimedia and Expo (ICME)*, 2021.

[51] D. Li, J. Zhang, and K. Huang, "Universal adversarial perturbations against object detection," *Pattern Recognition*, vol. 110, p. 107584, 2021.

[52] Y. Li, X. Bian, M.-C. Chang, and S. Lyu, "Exploring the vulnerability of single shot module in object detectors via imperceptible background patches," in *British Machine Vision Conference (BMVC)*, 2018.

[53] X. Wu, L. Huang, and C. Gao, "G-UAP: Generic universal adversarial perturbation that fools RPN-based detectors," in *Asian Conference on Machine Learning (ACML)*, 2019.

[54] X. Liu, H. Yang, Z. Liu, L. Song, Y. Chen, and H. H. Li, "DPATCH: An adversarial patch attack on object detectors," *AAAI Workshop on Artificial Intelligence Safety*, 2019.

[55] Y. Li, D. Tian, M. Chang, X. Bian, and S. Lyu, "Robust adversarial perturbation on deep proposal-based models," in *British Machine Vision Conference (BMVC)*, 2018.

[56] H. Zhang, W. Zhou, and H. Li, "Contextual adversarial attacks for object detection," in *IEEE International Conference on Multimedia and Expo (ICME)*, 2020.

[57] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[58] Y. Wang, T. yu an, W. Zhang, Y. Zhao, and X. Kuang, "An adversarial attack on DNN-based black-box object detectors," *Journal of Network and Computer Applications*, vol. 161, p. 102634, 03 2020.

[59] J. Lee and K.-i. Hwang, "Yolo with adaptive frame control for real-time object detection applications," *Multimedia Tools and Applications*, vol. 81, 2021.

[60] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[61] R. Taori, A. Dave, V. Shankar, N. Carlini, B. Recht, and L. Schmidt, "Measuring robustness to natural distribution shifts in image classification," in *Neural Information Processing Systems (NeurIPS)*, 2020.

[62] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, pp. 303–338, 06 2010.

[63] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[64] ——, "YOLOv3: An incremental improvement," *CoRR*, 2018. [Online]. Available: http://arxiv.org/abs/1804.02767

[65] J. C. Costa, T. Roxo, H. Proença, and P. R. M. Inácio, "How deep learning sees the world: A survey on adversarial attacks & defenses," *IEEE Access*, vol. 12, pp. 61 113–61 136, 2024.

[66] S. Y. Khamaiseh, D. Bagagem, A. Al-Alaj, M. Mancino, and H. W. Alomari, "Adversarial deep learning: A survey on adversarial attacks and defense mechanisms on image classification," *IEEE Access*, vol. 10, pp. 102 266–102 291, 2022.

[67] L. L. Ziyi Dong, Pengxu Wei, "Adversarially-aware robust object detector," in *European Conference on Computer Vision (ECCV)*, 2022.

[68] A. Saha, A. Subramanya, K. Patil, and H. Pirsiavash, "Role of spatial context in adversarial robustness for object detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020.

[69] P.-C. Chen, B.-H. Kung, and J.-C. Chen, "Class-aware robust adversarial training for object detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[70] H. Zhang and J. Wang, "Towards adversarially robust object detection," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[71] P.-y. Chiang, M. J. Curry, A. Abdelkader, A. Kumar, J. Dickerson, and T. Goldstein, "Detection as regression: certified object detection by median smoothing," in *Neural Information Processing Systems (NeurIPS)*, 2020.

[72] L. Zhou, Q. Liu, and S. Zhou, "Preprocessing-based adversarial defense for object detection via feature filtration," in *International Conference on Algorithms, Computing and Systems*, 2024.

[73] J. Liu, A. Levine, C. P. Lau, R. Chellappa, and S. Feizi, "Segment and complete: Defending object detectors against adversarial patch attacks with robust patch detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 14 973–14 982.

[74] T. Kim, Y. Yu, and Y. M. Ro, "Defending physical adversarial attack on object detection via adversarial patch-feature energy," in *ACM International Conference on Multimedia*, 2022.

[75] L. Strack, F. Waseda, H. H. Nguyen, Y. Zheng, and I. Echizen, "Defending against physical adversarial patch attacks on infrared human detection," in *IEEE International Conference on Image Processing (ICIP)*, 2024.

[76] N. Ji, Y. Feng, H. Xie, X. Xiang, and N. Liu, "Adversarial YOLO: Defense human detection patch attacks via detecting adversarial patches," *CoRR*, 2021. [Online]. Available: https://arxiv.org/abs/2103.08860

[77] C. Xiang and P. Mittal, "DetectorGuard: Provably securing object detectors against localized patch hiding attacks," in *ACM SIGSAC Conference on Computer and Communications Security*, 2021.

[78] K.-H. Chow, L. Liu, M. E. Gursoy, S. Truex, W. Wei, and Y. Wu, "Understanding object detection through an adversarial lens," in *European Symposium on Research in Computer Security*. Springer, 2020, pp. 460–481.

[79] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations*, 2018.

[80] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, , Q. Jiang, C. Li, J. Yang, H. Su, J. Zhu, and L. Zhang, "Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection," in *European Conference on Computer Vision (ECCV)*, 2024.

[81] L. H. Li, P. Zhang, H. Zhang, J. Yang, C. Li, Y. Zhong, L. Wang, L. Yuan, L. Zhang, J.-N. Hwang, K.-W. Chang, and J. Gao, "Grounded language-image pre-training," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[82] Y. Bai, J. Mei, A. Yuille, and C. Xie, "Are transformers more robust than CNNs?" in *Neural Information Processing Systems (NeurIPS)*, 2021.

[83] S. Bhojanapalli, A. Chakrabarti, D. Glasner, D. Li, T. Unterthiner, and A. Veit, "Understanding robustness of transformers for image classification," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[84] D. Zhou, Z. Yu, E. Xie, C. Xiao, A. Anandkumar, J. Feng, and J. M. Alvarez, "Understanding the robustness in vision transformers," in *International Conference on Machine Learning*, 2022.

[85] L. Engstrom, B. Tran, D. Tsipras, L. Schmidt, and A. Madry, "Exploring the landscape of spatial robustness," in *International Conference on Machine Learning (ICML)*, 2019.

[86] A. S. Shamsabadi, R. Sánchez-Matilla, and A. Cavallaro, "ColorFool: Semantic adversarial colorization," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[87] R. Duan, X. Ma, Y. Wang, J. Bailey, A. K. Qin, and Y. Yang, "Adversarial camouflage: Hiding physical-world attacks with natural styles," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.