International Journal of Computer Vision
https://doi.org/10.1007/s11263-024-02096-6

Benchmarking Object Detection Robustness against Real-World Corruptions

Jiawei Liu[1] · Zhijie Wang[2] · Lei Ma[2,3] · Chunrong Fang[1] · Tongtong Bai[1] · Xufan Zhang[1] · Jia Liu[1] · Zhenyu Chen[1]

Received: 3 December 2022 / Accepted: 23 April 2024

Abstract

With the rapid recent development, deep learning based object detection techniques have been applied to various real-world software systems, especially in safety-critical applications like autonomous driving. However, few studies are conducted to systematically investigate the robustness of state-of-the-art object detection techniques against real-world image corruptions and yet few benchmarks of object detection methods in terms of robustness are publicly available. To bridge this gap, we initiate to create a public benchmark of COCO-C and BDD100K-C, composed of sixteen real-world corruptions according to the real damages in camera sensors and image pipeline. Based on that, we further perform a systematic empirical study and evaluation of twelve representative object detectors covering three different categories of architectures (*i.e.*, two-stage, one-stage, transformer architectures) to identify the current challenges and explore future opportunities. Our key findings include (1) the proposed real-world corruptions pose a threat to object detectors, especially for the corruptions involving colour changes, (2) a detector with a high mAP may still be vulnerable to real-world corruptions, (3) if there are potential cross-scenarios applications, the one-stage detectors are recommended, (4) when object detection architectures suffer from real-world corruptions, the effectiveness of existing robustness enhancement methods is limited, and (5) two-stage and one-stage object detection architectures are more likely to miss detect objects compared with transformer-based methods against the proposed corruptions. Our results highlight the need for designing robust object detection methods against real-world corruption and the need for more effective robustness enhancement methods for existing object detectors.

Keywords Computer vision · Robustness · Object detection · Data augmentation

1 Introduction

Object detection refers to the technique that determines whether there are any instances of objects from a predefined set of categories in a given image and where they are located (Liu et al., 2020). Compared to image classification, object detection additionally enables the user to know the coordinates of objects precisely by localization. The rapid development of deep learning techniques has greatly improved the accuracy of object detection techniques (Pouyanfar et al., 2019), enabling the option to solve more complex real-world tasks with object detection. Nowadays, object detection has been widely deployed in diverse commercial and industrial domains and system applications, including robot vision (She et al., 2020), autonomous driving (Garcia et al., 2020), augmented reality (Liu et al., 2019b), *etc*.

Communicated by Oliver Zendel.

Jiawei Liu and Zhijie Wang have contributed equally to this work.

✉ Chunrong Fang
fangchunrong@nju.edu.cn

✉ Jia Liu
liujia@nju.edu.cn

✉ Zhenyu Chen
zychen@nju.edu.cn

[1] State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China

[2] University of Alberta and Alberta Machine Intelligence Institute (Amii), Edmonton, Canada

[3] The University of Tokyo, Tokyo, Japan

Published online: 15 May 2024

Despite the rapid development of deep learning-based object detection techniques, concerns are also raised about their robustness and reliability since they might be used in safety-critical applications like autonomous driving. A small flaw in the object detector of autonomous driving systems could result in severe consequences. For instance, even images with visually imperceptible perturbations, i.e., the adversarial examples, may cause the failures of object detectors (Xie et al., 2017). In the real-world scenario, a missed detection of Tesla runs into a car crash (Times, 2017; CNN, 2016). While among the root causes that trigger the flaws in object detectors, one of the major causes could be the corrupted data (Hendrycks et al., 2019). However, there is a lack of comprehensive studies on object detection robustness against real-world corruptions. Most of the existing studies have been using common corruptions to benchmark image classification and semantic segmentation (Hendrycks et al., 2019; Kamann and Rother, 2021). Michaelis et al. (2019) conduct a benchmark to assess object detection's performance by combining common corruptions and real-world weather corruptions, whereas these common corruption patterns might either be unreal or seldom happens in real-world environments. Given the fact that images used by object detectors are usually directly captured from cameras without any manual processing, damages of camera sensors or image processing could bring several kinds of real-world corruptions (Garcia et al., 2020). However, it is still unclear to what extent the performance of existing object detectors would be degraded against such real-world corruptions. As object detectors become a key component in many real-world intelligent software systems, it is of great importance to systematically investigate their potential risks and limitations during real-world usage.

To bridge this gap, in this paper, we initiate an early step and present the first benchmark for object detection against real-world corruptions, and perform a systematic analysis to benchmark the performance of representative object detection methods and robustness enhancement techniques. The real-world corruptions are designed according to the real damages in image sensing pipeline of camera sensors for object detection software systems, which ensure the performance of higher-level algorithms (Schwartz et al., 2019). Our high-level study workflow is summarized in Fig. 4. In particular, we mainly investigate research questions from five important perspectives, to identify the challenges and potential opportunities for building safe and reliable software systems based on object detection.

In summary, this paper makes the following contributions:

- We design the first series of real-world corruptions based on the real damages that could occur in image pipeline of camera sensors.
- We create the first publicly available benchmark of object detection with different kinds of architectures that span over various industrial domains, which initiates a very early step and enables many potential follow-up research along this direction.
- We perform a systematic analysis of the performance (*i.e.*, error rates and flaw symptoms) of the selected object detection methods and robustness enhancement methods.
- Based on the analysis results, we further pose discussions on the challenge of future directions on building robust object detection, including comprehensive evaluation metrics, improving bounding box localization and avoiding missed detection flaws when encountering real-world corruptions.

To the best of our knowledge, this is the very first paper that establishes a publicly available dataset and benchmark for diverse categories (two-stage, one-stage, transformer architectures) of object detection methods against real-world corruptions.[1] The benchmark and our study results demonstrate the potential research opportunities around object detection techniques to meet the growing demands for robust object detection. Our work enables better understanding, establishes the basis, and paves the path toward further quality assurance research to build safer and more reliable software systems.

2 Background and Related Work

2.1 Object Detection

Overview. Object detection has been employed for detecting the existence of objects in a given image and returning the spatial location and extent of each instance. As an important task in the computer vision domain, object detection has become a key component of many intelligent software systems and has been employed in many real-world application scenarios (Pathak et al., 2018), making the systematic study on which very important. For example, in a safety-critical system, *e.g.*, autonomous driving (Tian et al., 2018), the system would require not only what classifications of objects are present in the current field of view, but also, more importantly, where these objects are located (Feng et al., 2021) to avoid collisions. The same requirements are also posted in other real-world applications such as medical diagnosis (Shen et al., 2017) and intensive crowd detection (Sindagi and Patel, 2018).

Definition To be specific, object detection is aimed at locating object instances from a large number of predefined

[1] https://sites.google.com/view/real-worldbenchmark.

categories in images (Liu et al., 2020). The detection is mainly conducted by a detector $\mathbb{D}$, which is defined as follows:

Definition 1 (*Detector*) In a given scenario, an object detector $\mathbb{D}$ is expected to determine whether there are instances of objects $O_0, \ldots, O_n$ in the detected image I from predefined categories $C_0, \ldots, C_m$, and, if present, to return a bounding box B of each instance. A bounding box B is defined by its spatial location (x, y) with its width w and height h.

$$\mathbb{D}(I) = \{O_0[C_i, B(x, y, w, h)], \ldots\} \tag{1}$$

where $0 \leq i \leq m$, and (x, y) may denote the center or the upper-left corner of the bounding box.

Since the task of object detection involves the two subtasks of object localization and classification, the approaches based on deep learning for object detection mostly fall into three main categories:

- Two-stage detection, also called region-based detection, employs a pre-processing stage for generating object proposals, including (Szegedy et al., 2013; Sermanet et al., 2014; Erhan et al., 2014a, b).
- One-stage detection, or region proposal free detection, has a single proposed method which does not separate the process of the detection proposal, including (Redmon et al.., 2016; Liu et al., 2016; Bolya et al., 2019; Duan et al., 2019).
- Transformer detection treats the object detection as a prediction problem for a collection, using the encoder-decoder structure, including (Carion et al., 2020; Zhu et al., 2021).

Methodologies. In the early ages, the study of object detection is based on template matching techniques and simple part-based models. Fischler and Elschlager (1973) propose a method to find a visual object with descriptions. Since then, handcrafted local invariant features begin to be popular, such as the Scale Invariant Feature Transform (SIFT) proposed by Lowe et al. (1999). Since 2014, researchers have started to adopt deep learning techniques for object detection. Girshick et al. propose the two-stage detector, RCNN, which obtains record-breaking results in the detection of general object categories (Erhan et al., 2014b). To reduce the computational cost for current mobile/wearable devices, Liu et al. (2016) propose the one-stage detector, SSD. In addition to using CNNs, recent work has also leveraged transformer-based architecture to solve object detection problems, including DETR, a transformer detector proposed by Carion et al. (2020). Many of these object detection methods have also been involved in solving complex realistic problems in real-world scenarios (Litjens et al., 2017), *e.g.*, autonomous driving, intelligent video surveillance, and augmented reality. Wu et al. propose Squeezedet, a fully convolutional neural network for autonomous driving with real-time inference speed, small model and energy efficiency (Wu et al., 2017). Liu et al. (2022) employ Faster R-CNN for object detection in medical images via the additive secret sharing technique and edge computing.

2.2 Object Detection Robustness

Overview. The robustness of object detection indicates the degree to which a detector can perform correctly in the presence of invalid inputs or stressful environmental scenarios (Shekar et al., 2021). Several recent efforts demonstrate that the object detectors are suffering from the threat of non-robustness (Islam et al., 2019; Sun et al., 2015). In real-world applications, the object detectors must be robust enough to cope with various challenges and uncertainties. For example, in the autonomous driving scenario, an ideal object detector is required to maintain its reliability under various weather conditions, different road conditions, and may even be required to face hardware damages. Therefore, a lot of work has been devoted to the study of object detection robustness.

Datasets. Currently, several datasets have been made publicly available for evaluating robustness. Dan et al. propose a dataset named ImageNet-C with 15 common corruptions from noise, blur, weather, and digital categories to measure corruption robustness and ImageNet-P with perturbation sequences to measure perturbation robustness (Hendrycks et al., 2019). Michaelis et al. use 15 common corruptions of ImageNet-C and generate new datasets, i.e., PASCAL-C and Cityscapes-C for robustness evaluation (Michaelis et al., 2019). These public datasets are constructed with common corruptions. However, to evaluate the robustness of object detectors in real-world scenarios, except for common corruptions, the robustness datasets need to include additional corruptions corresponding to real-world situations.

Methodologies. Many methodologies have been proposed to generate augmented data and adversarial samples for the study of object detection robustness. The test-time data augmentation on accuracy is a well-known mechanism for measuring the robustness (Shorten and Khoshgoftaar, 2019). Minh et al. (2018) measure robustness by distorting test images with a 50% probability and contrasting the accuracy on un-augmented data with the augmented data. Carlson et al. propose an efficient, automatic, physically-based augmentation pipeline and improve the robustness of object detection in urban driving scenes. Zhong et al. (2020) obtain reasonable improvement on the robustness object detection with the data augmentation method. Rebuffi et al. (2021) combine the model weight averaging with data augmentation to improve

robustness. Meanwhile, many studies exist on investigating adversarial robustness. Sobh et al. (2021) apply both white and black box adversarial attacks on object detectors to evaluate robustness. Zhang et al. (2022) propose a model-agnostic adversarial defensive method and improve the robustness of object detectors. Kim et al. (2021) propose an optical adversarial attack to evaluate the robustness of object detectors. Both data augmentation and adversarial attack methodologies show promising achievements in evaluating and improving robustness. However, these methodologies may also have the potential to produce data that does not match real-world scenarios.

Benchmarks. Up to now, several public benchmarks on object detection and robustness have been proposed. These benchmarks show the performance of object detectors regarding accuracy and robustness. Chen et al. (2019b) propose an object detection toolbox named MMDetection, which contains a rich set of object detectors with related components and modules, as well as the accuracy of the detectors. Michaelis et al. (2019) provide an easy-to-use benchmark to assess the robustness of object detectors. The Robust Vision Challenge[2] is proposed to measure the performance of object detectors across several challenging benchmarks with different characteristics, e.g., indoors versus outdoors, real versus synthetic, sunny versus bad weather, and different sensors. Kamann and Rother (2021) present a robustness benchmark with DeepLabv3+ implemented on various network backbones. Dong et al. (2020) propose a comprehensive and coherent benchmark to evaluate adversarial robustness. These benchmarks provide considerable insights for the research of object detection and robustness. Nowadays, object detectors have been increasingly adopted in real-world applications. More benchmarks concerning object detection robustness in real-world scenarios are necessary for further research in multiple domains.

Figure 1 shows the landscape of the above-mentioned interrelated aspects. Up to now, lots of studies have been proposed regarding object detection and robustness. Different from existing studies regarding common corruptions, adversarial attacks and augmentations, we focus on the robustness of object detection against real-world corruptions. To this end, we aim to propose new datasets with corruptions corresponding to real-world situations, and present a new benchmark reflecting the object detection robustness in real-world scenarios.

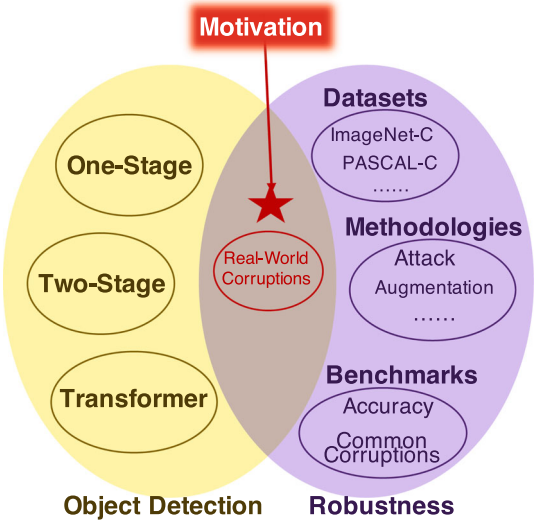

Fig. 1 The landscape of interrelated aspects and our motivation

2.3 Image Sensing Pipeline of Cameras

In applications like autonomous driving (Garcia et al., 2020) and video surveillance (Elharrouss et al., 2021), object detection usually requires real-time processing. Therefore, the images collected by the sensors, *i.e.*, cameras, will be sent to object detectors directly without manual quality enhancement. However, these images could be damaged in different stages of the camera's (image signal processor (ISP), resulting in incorrect detection results, and even crashes. ISP is an important hardware in cameras dedicated to image processing tasks, ensuring the performance of higher-level algorithms (Schwartz et al., 2019). A typical ISP of a camera in a real-world autonomous driving scenario is shown in Fig. 2. In such a scenario, the camera receives reflected light from the real world and eventually converts the light signal into the electrical signal (Kawamura, 1998). The images from the electrical signal are then used by the autonomous driving system for object detection.

Within the internal components of the camera, the *lens group* is used to receive the reflected light from the real world and pass the light onto the sensor chip (Fossum, 1997). The *sensor chip* makes use of the Bayer pattern to convert the light signal into raw data of the digital signal, *i.e.*, the RAW image (Liu et al., 2019a). Since the Bayer pattern places single colour filters on each pixel at intervals, the RAW image yields a partially voided image in R, G, and B channels, as shown in Fig. 3. To convert a RAW image into a JPG image that can be directly used for the inference of downstream tasks, the *image signal processor (ISP)* (Zhou et al., 2007) then transforms the image automatically. Figure 3 shows a simplified pipeline of digital image processing functions within the ISP, consisting of mandatory basic functions and optional enhancement functions.

Black level correction (BLC) counteracts hidden electric current disturbances by providing gains in the R and G channels for RAW data. Lens shading correction (LSC) is applied with colour casts to correct the shading caused by convex

[2] http://www.robustvision.net/.

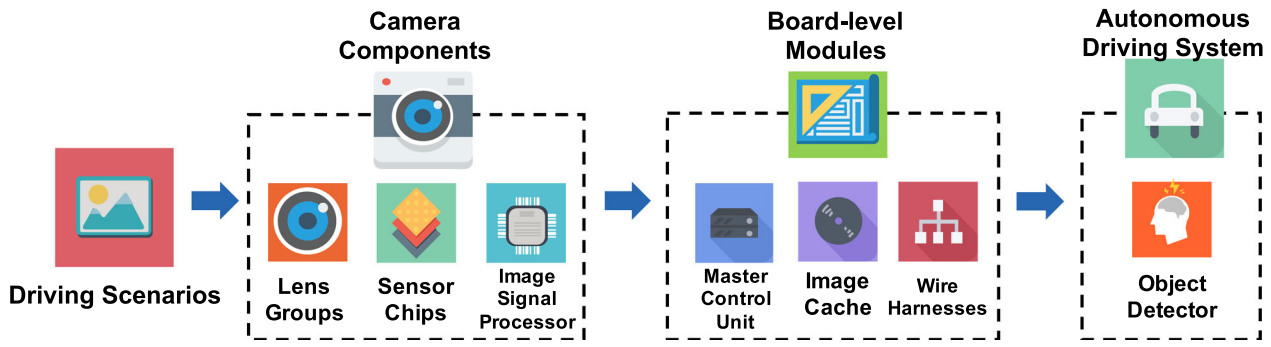

Fig. 2 Image pipeline in a real-world scenario

lens lenses. Automated white balance (AWB) applies different gains to RGB components and compensate for colour differences in terms of the illuminant. Bad pixel correction (BPC) is responsible for the removal of dead points in the conversion of the light signal. Colour filter array interpolation (CFA-I) interpolates the two missing colour components at each pixel and produces the JPG image. Gamma correction (GC) corrects the linear relationship between current and luminance to a non-linear relationship that is consistent with human perception. Colour space conversion (CSC) converts images from RGB colour space to YUV one to reduce image noises.

The components on the *Board-level Modules* cooperate to input the images into the Autonomous Driving System. *Master Control Unit* provides the power and driving clock for the image sensor chip. *Image Cache* converts the data into a streaming format. *Wire Harness* transfers the streaming to the Autonomous Driving System. Each component is closely integrated, and a noise pattern in any one of them could lead to flaws that threaten the Autonomous Driving System.

3 Study Design

3.1 Overview

As illustrated in Fig. 4, we perform our study to investigate five research questions. In RQ1, we evaluate the effectiveness of real-world image corruptions proposed in this paper. In RQ2, we evaluate the performance of popular object detectors against the proposed corruption patterns. In RQ4, we investigate the robustness of object detectors by a cross-scenarios evaluation. In RQ4, we evaluate the effectiveness of enhancement methods for object detection. In RQ5, we investigate the flaw symptoms found in different object detection methods throughout our evaluation. In this section, we first introduce the design of our real-world corruption patterns in Sect. 3.2, and benchmark design in Sect. 3.3. Then we detail the design of our research questions in Sect. 3.5. In this study,

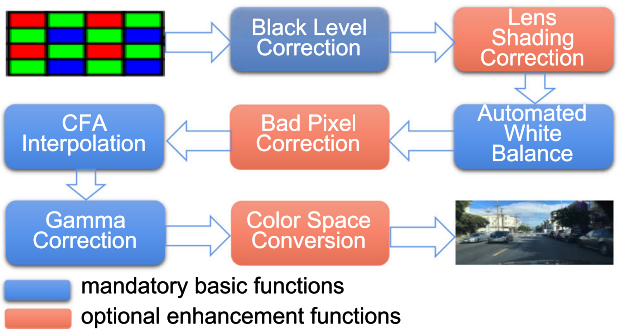

Fig. 3 Image pipeline functions in a typical image signal processor

we would construct an object detection benchmark to enable systematic studies and further research along this direction to enhance the object detection component in an intelligent software system.

3.2 Real-World Corruptions

As discussed in Sect. 2, object detection has been used in many safety-critical domains, *e.g.* autonomous vehicle systems, and noise patterns produced in any stage of camera's image pipeline could result in flaws of the object detection for autonomous driving. Extensive studies have shown the common corruption patterns could significantly degrade the performance of image classifiers (Hendrycks et al., 2019), however, there has not been any systematic study on the effects of real-world corruptions that: (1) follow image pipeline of the camera, (2) occur within the process of camera's internal processing. To bridge this gap, in this paper we propose 16 real-world image corruption patterns based on the potential damages that occur in different stages of camera's image pipeline. We group these 16 corruption patterns into three categories according to different locations of damages: (1) camera damage, (2) ISP damage, and (3) board-level damage.

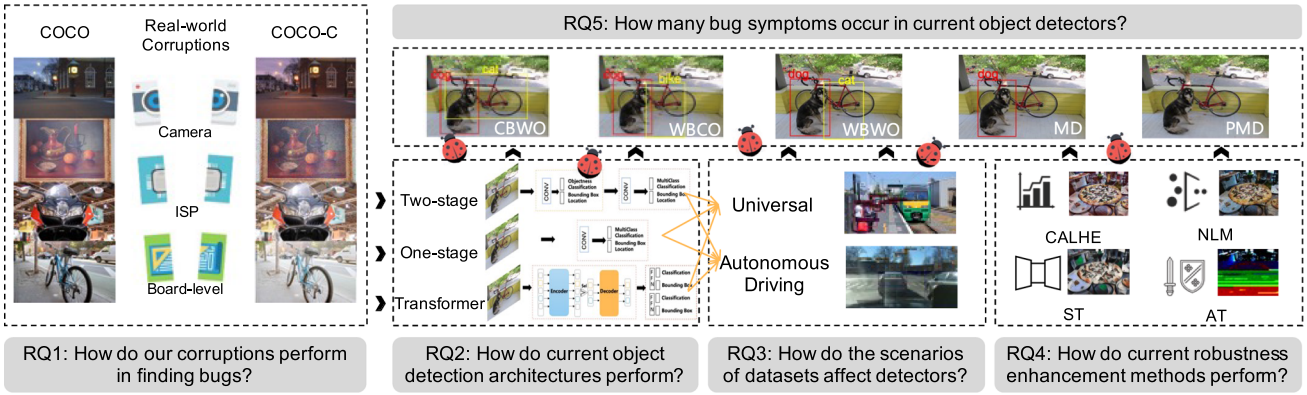

Fig. 4 Study workflow and research questions

3.2.1 Camera Damage Corruptions

Among the camera components, the common real-world corruptions on *Lens Group* are *Fog*, *Lens Obstruction*, *Focus Motor Damage* and on *Sensor Chips* are *CCD Sensor Damage*, *CMOS Sensor Damage*.

Fog (F). The corruption F leads to scattering of the reflected light received by the camera. According to the atmospheric scattering formulation, the center of fogging, the scattering coefficient, the transmittance and the atmospheric light intensity can be obtained to reproduce this corruption. The atmospheric scattering formulation is formulated according to the existing method(Bruneton and Neyret, 2008).

Lens Obstruction (LO). The corruption LO obscures the reflected light. We formulate this corruption by adding obscuring drops and flipping the reflected light in the obscured area. The formulation of corruption LO is reproduced with DirtyGAN (Uricar et al., 2021).

Focus Motor Damage (FM-D). The corruption FM-D makes the reflected light out of focus. We formulate this corruption by adding unfocused noise. The unfocused noise of FM-D is generated with motion blur model (Lin et al., 2012).

CCD Sensor Damage (CCD-D). The corruption CCD-D causes white dot and line damage when the camera converts the light signal into a digital signal. We formulate this corruption by adding white dots and line damage to the digital signal, which is reproduced by the generic model in Antilogus et al. (2014).

CMOS Sensor Damage (CMOS-D). The corruption CCD-D causes white dot and black line damage on the digital signal. We formulate this corruption by adding black line damage to the digital signal, which is reproduced according to the laser spot model (Ying et al., 2009).

3.2.2 ISP Damage Corruptions

As the complicated component in the camera, the ISP could be involved with diversified real-world corruptions in the pipeline functions.

Insufficient Black Level Correction (I-BLC) and Excessive Black Level Correction (E-BLC). The corruption I-BLC and E-BLC lead to improper gains on images by BLC. We formulate the insufficient corruption by adding insufficient gains on channel *G* and the excessive corruption by adding excessive gains on channel *G*. These insufficient gains are reproduced according to the algorithms in Zhou et al. (2007).

Lens Shading Correction Damage (LSC-D). The corruption LSC-D causes loss of brightness in the image corners. We formulate this corruption with gradual decay of brightness from the center of the image to the corners. The decay is reproduced according to the algorithms in Silva et al. (2016).

Automated White Balance Damage (AWB-D). The corruption AWB-D leads to the failure to compensate for illuminant differences in images. We formulate this corruption by applying different gains to RGB channels. The gains are reproduced according to the algorithms in Zhou et al. (2007).

Bad Pixel Correction Damage (BPC-D). The corruption BPC-D causes some of the pixels to be bad dots with incorrect information. We formulate this corruption by adding black pixels as bad pixels. The black pixels are reproduced according to the algorithms in Celestre et al. (2016).

Colour Filter Array Interpolation Damage (CFAI-D). The corruption CFAI-D leads to a partial loss of channel information when converting Bayer RG images to RAW images. We formulate this corruption by dropping channel information partially on Bayer RG images. The dropping is reproduced according to the algorithms in Zhou et al. (2007).

Gamma Correction Damage (GC-D). The corruption GC-D causes abnormal brightness distribution of the image.

We formulate this corruption by modifying the contrast of the brightness. The contrast modification is reproduced according to the adaptive gamma correction method in Rahman et al. (2016).

Colour Space Conversion Damage (CSC-D). The corruption CSC-D leads to the failure of colour noise removal and edge enhancement on the YUV colour space. We formulate this corruption by adding colour noise and blurring the edge on the YUV colour space, which is reproduced according to the YUV model in Chaves-González et al. (2010).

3.2.3 Board-level Damage Corruptions

The corruptions belonging to this category indicate real-world damages that occur on the components of the board-level module.

Synchronization Exceptions (SE). The corruption SE causes the image data with exceptions in the time sequence when it is imported into the system. We formulate this corruption by dropping image data in the time sequence. The dropping in the time sequence is reproduced chaotic synchronization phenomena in Volos et al. (2013).

Memory Exceptions (ME). The corruption ME leads to missing data in the memory. We formulate this corruption by dropping data in the memory. The dropping in memory is reproduced according to the image encoding steps in Guo et al. (2016).

Transfer Harness Exceptions (THE). The corruption THE causes damage to some of the channels when the image data is transferred in YUV format. We formulate this corruption by dropping the information in some of the channels of YUV format, which is reproduced according to the image wireless transfer model in Chandra et al. (2016).

3.3 Benchmark Datasets

To benchmark the performance of popular object detection methods against these corruption patterns, we make use of the COCO dataset and the BDD100K dataset. COCO is a large-scale object detection dataset with more than 200,000 images and 80 object categories (Lin et al., 2014). Data from COCO are common objects in society and natural context, rendering a perfectly clean dataset. The images in BDD100K are the frames at the 10th second in the videos in a diverse driving dataset for heterogeneous multitask learning. To this end, we construct COCO-C and BDD100K-C based on our proposed corruption patterns. It is worth noting that our devised corruptions are general to be applied to other datasets, e.g., KITTI.

Setup. We create COCO-C and BDD100K-C with five severity levels for each corruption, which is the same as ImageNet-C. Figure 5 illustrates samples from COCO-C with severity level 5, and they clearly still preserve the semantics

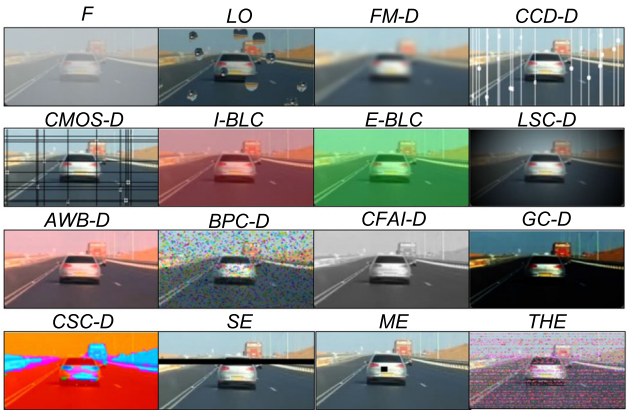

Fig. 5 Visualizations of COCO-C with severity level 5

of the objects. These designed corruptions are applied to the validation set of COCO and BDD100K, resulting in COCO-C and BDD100K-C— two 80 times larger datasets to test the robustness of object detection methods.

Metrics. The commonly used metric for evaluating object detectors are *mean Average Precision* (mAP) and Intersection over Union (IoU). mAP demonstrates how accurate a detector is on detecting objects. IoU is the geometric overlap ratio between two bounding boxes, to measure how accurate a detector is on locating accuracy objects. In this work, however, we are more concerned with how badly they are wrong than how well they are correct , i.e., to what extent a detector makes errors on locating objects. Therefore, we propose new metrics: *Corruption Error* (CE) and *mean Corruption Error* (mCE) to evaluate the robustness of a given object detection method.

As shown in Fig. 7, for a given object detector d, suppose the ground-truth bounding box of an object O_0 is defined by a pair of coordinates $((x_2, y_2), (x_3, y_3))$, and the predicted bounding box is defined by $((x_1, y_1), (x_4, y_4))$, then we denote the error rate on this object as

$$err^d_{O_0} = 1 - \frac{|x_4 - x_3| * |y_3 - y_4|}{|x_2 - x_1| * |y_2 - y_1|} \tag{2}$$

Thus, CE is calculated by the ratio of the non-overlap region between the ground-truth bounding box and the predicted bounding box. With CE, we could provide a more intuitive characterization on the flaws of the object detectors over their robustness. Further, for a image I with n different objects, we denote its error rate err^d_I as $\frac{\sum_{i=0}^{n-1}(err^d_{O_i})}{n}$. For a clean dataset, *i.e.*, the one without applying any corruption patterns, we denote the average error rate on this dataset as E^d_{clean}.

Real-world corruptions such as *CMOS-D* can be benign or destructive depending on their severity. Figure 6 shows an

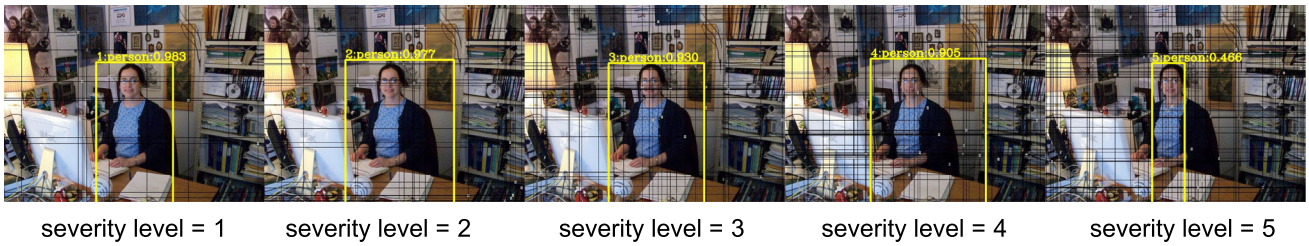

Fig. 6 Predicted bounding boxes from *SSD* (Liu et al., 2016) on an image corrupted by different severity levels of *CMOS-D*. The higher the severity levels, the greater the threats of corruptions to robustness

example on the change of predicted bounding box of *SSD* against *CMOS-D* corruption on different severity levels. In order to comprehensively evaluate a detector's robustness against a given certain type of corruption, we aggregate the detector's error rate across five corruption severity levels and propose *Corruption Error* (CE) as the following.

$$CE_c^d = \frac{\sum_{s=1}^{5}(E_{i,c}^d)}{5} \tag{3}$$

where c denotes a certain corruption pattern, and s denotes the severity level.

In addition to CE, *mean Corruption Error* (mCE) is proposed to evaluate a detector's robustness against all corruptions.

$$mCE^d = \frac{\sum_c CE_c^d}{|C|} \tag{4}$$

where C is a set of real-world corruption patterns, and $|C|$ denotes the size of C, which is 16 in this paper.

We use CE and mCE as the main metrics for benchmarking. We will release our leaderboard publicly to facilitate future studies on performance of object detectors.

3.4 Object Detectors

To construct a comprehensive and systematic benchmark, in this study we select twelve representative object detection methods (as shown in Table 1) according to: (1) performance on original COCO and BDD100K dataset, and (2) diverse model architecture. Each of them marks a milestone contribution to the development of deep learning-based object detection methods.

Two-stage detectors. The two-stage detectors generate category-independent region proposals for objects in a given image, and extract features from these regions to determine the category labels of the objects. Among the two-stage detectors, we select Faster R-CNN (Ren et al., 2015), Mask R-CNN (He et al., 2017), RetinaNet (Lin et al., 2017), Cascade R-CNN (Cai and Vasconcelos, 2019), Cascade Mask R-CNN (Cai and Vasconcelos, 2019), and Hybrid Task Cascade (Chen et al., 2019a). *Faster RCNN* first employs an accurate region proposal network for generating region proposals efficiently. *Mask R-CNN* adds a fully convolutional network to get a binary mask for each region in parallel to predicting the object label. *RetinaNet* reshapes the standard cross entropy loss to address the problem of foreground-background class imbalance. *Cascade R-CNN* adopts end-to-end learning of more than two cascaded classifiers for generic object detection. *Cascade Mask R-CNN* trains detectors sequentially to avoid overfitting, using the output of the former as training set for the next. *Hybrid Task Cascade* learns discriminative features progressively while integrating complementary features in each cascade.

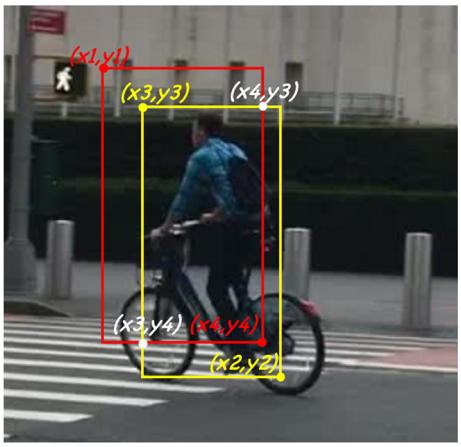

Fig. 7 The detected (red) and expected (yellow) bounding box of the object

One-stage detectors. To reduce computational costs, the one-stage detectors encapsulate all computation in a single network by directly predicting class probabilities and bounding box offsets from full images, without region proposal generation. Among the one-stage detectors, we select YOLO v3 (Redmon and Farhadi, 2018), SSD (Liu et al., 2016), YOLACT (Bolya et al., 2019), and CenterNet (Duan et al., 2019). *YOLO v3* predicts an objectness score for each bounding box using logistic regression and achieves obvi-

Table 1 Selected object detectors in the benchmark

Category	Year	Method	Detector
Two-stage	2015	Faster R-CNN (Ren et al., 2015)	FRC
	2017	Mask R-CNN (He et al., 2017)	MRC
	2017	RetinaNet (Lin et al., 2017)	RN
	2018	Cascade R-CNN (Cai and Vasconcelos, 2019)	CRC
	2019	Cascade Mask R-CNN (Cai and Vasconcelos, 2019)	CMRC
	2019	Hybrid Task Cascade (Chen et al., 2019a)	HTC
One-stage	2016	SSD (Liu et al., 2016)	SSD
	2018	YOLO v3 (Redmon and Farhadi, 2018)	YL3
	2019	YOLACT (Bolya et al., 2019)	YLA
	2019	CenterNet (Duan et al., 2019)	CN
Transformer	2020	DETR (Carion et al., 2020)	DETR
	2021	Deformable DETR (Zhu et al., 2021)	DDETR

ous improvements in efficiency. *SSD* performs detection over multiple scales by operating on multiple feature maps to preserve real-time speed without sacrificing too much detection accuracy. *YOLACT* further improves the efficiency by generating a set of prototype masks and predicting per-instance mask coefficients. *CenterNet* models an object as a single point to avoid the exhaustive list of potential object locations.

Transformer-based detectors. The recent progress has shown that transformer-based architecture, which is originally proposed for NLP tasks and also shown to be effective for CV applications. Different from the above local-to-global detection methods, transformers are global-to-local detection methods. We select DETR (Carion et al., 2020) and Deformable DETR (Zhu et al., 2021) as representative transformer-based detectors. *DETR* implements object detection as a direct set prediction problem via bipartite matching, leveraging a transformer encoder-decoder architecture. *Deformable DETR* employs attention modules to avoid slow convergence and limited feature spatial resolution.

3.5 Research Questions

RQ1: How do our corruptions perform in finding flaws?

RQ1 aims to study the effectiveness of the proposed 16 corruptions in two dimensions. On the one hand, these corruptions should effectively expose the vulnerabilities of the object detection models. To this end, we leverage twelve representative object detection methods introduced in Sect. 3.4. For a fair benchmark, we re-train each object detection method on the clean dataset, *i.e.*, the original training data of COCO and BDD100K, and adopt the same training strategy as described in each method's paper and documentation. We first report the CE of each object detector on the clean dataset. Then, we calculate the CE of each detector on different corruption and categorized the results into three categories according to the location of damaged camera components (*i.e.*, camera damage, ISP damage and board-level damage). By analyzing the evaluation results, we'd like to evaluate whether our proposed corruption patterns can find flaws and limitations of popular object detection methods.

On the other hand, the data generated by these corruptions should be sufficiently realistic. We employ structural similarity (SSIM) (Wang et al., 2004), information fidelity criterion (VIF) (Sheikh and Bovik, 2006) and visual information fidelity (IFC) (Sheikh et al., 2005) to measure the realism of the generated data. SSIM is a benchmark criterion for evaluating the structural information of generated data. VIF measures the information fidelity of generated data. IFC measures the visual quality of generated data. The higher the evaluation results are, the better the realism of the corruption is.

RQ2: How do current object detectors perform against real-world corruptions?

For RQ2, we focus on assessing the existing object detector's robustness by interpreting their average mCE on COCO-C and BDD100K-C w.r.t. different model architectures. We first divide the twelve object detection methods into three categories according to their model architectures: (1) two-stage architectures, (2) one-stage architectures, and (3) transformer-based architectures. In the experiment, we calculate mCE of each architecture and observe their performance by category. By observing these results, we'd like to learn more about the characters of model architectures when designing a robust object detector against real-world image corruptions.

RQ3: How do the scenarios of datasets affect detectors?

In different application scenarios, the data used by object detection models can vary. In a universal scenario, there are various objects to be detected with a low level of density, e.g. in shopping software to identify the category of goods. In the autonomous driving scenario, there is limited variety but high

density of objects to be detected, e.g. detecting obstacles on the road. What are the consequences when an object detection model uses data that does not fit the use scenario? Are the consequences consistent across different models? In RQ3, we will investigate these questions.

Currently, object detection models are widely used in universal scenarios, but also play an important role in the autonomous driving scenarios. Therefore, to answer RQ3, we train all models with data from universal (*i.e.*, COCO) and autonomous driving scenarios (*i.e.*, BDD100K), respectively. To investigate the consequences of data not matching the scenarios, we evaluate the mCE of autonomous driving models with data from the universal scenario (*i.e.*, COCO-C) and the universal models with data from autonomous driving scenario (*i.e.*, BDD100K-C).

RQ4: How do current robustness enhancement methods perform against real-world corruptions?

For RQ4, we focus on investigating the effectiveness of existing robustness enhancement methods which have been proved to be effective in improving performance on many computer vision tasks (Hendrycks et al., 2019). However, it is still unclear whether these methods would be effective for enhancing the detector's robustness against real-world corruption patterns. In the experiment, we make use of four representative enhancement strategies. Specifically, we leverage Contrast Limited Adaptive Histogram Equalization (CLAHE) (Pizer et al., 1987), Style Transfer (ST) (Geirhos et al., 2019), Non-Local Means (NLM) (Buades et al., 2005) and Adversarial Training (AT) (Tramèr et al., 2018) as enhancement strategies. CLAHE increases the contrast of an image by local histogram equalization, making the objects easier to be detected and reduce the effect of some corruption patterns. ST extends the sample space of object detectors and increases the diversity of the image by transferring the image across different styles. With ST, the generalization of detectors could be improved and leading to high robustness.

NLM improves the images in terms of quality and clarity by reducing the noises, which helps the detectors to identify the objects. AT improves the robustness of object detectors to noise, perturbation and adversarial attacks, by minimizing the worst-case error when the data is perturbed by an adversary. We adopt the original hyper-parameter settings which are consistent with their official implementations in our study. We calculate CE for each of the different detectors after deploying one of three techniques. By comparing the detector's performance before/after enhancement, we'd like to answer whether these enhancement techniques are still effective against real-world corruptions.

RQ5: How many flaw symptoms occur in current object detectors?

There are multiple flaw symptoms in object detectors, such as build errors, launch errors, and logic errors (Garcia et al., 2020). Since the architectures adopted in this paper are consistent with their official implementations, we avoid flaw symptoms like build errors. Therefore, in RQ5, we aim to investigate the flaw symptoms independent of external factors. To better understand the flaws revealed in RQ2 - RQ4, we perform exploration on five typical flaw symptoms divided by flaws on different output's component of object detectors (defined in Definition 1) as follows:

- Correct bounding box with wrong objects (**CBWO**): The detector locates the object in the correct bounding box,[3] but classifies the object into a wrong category.
- Wrong bounding box with correct objects (**WBCO**): The detector locates the object in the wrong bounding box but classifies the object into a correct category.
- Wrong bounding box with wrong objects (**WBWO**): The detector locates the object in the wrong bounding box and classifies the object into a wrong category.
- Partial missed detection (**PMD**): The detector fails to detect some of the objects in an image.
- Missed detection (**MD**): The detector fails to detect all of the objects in an image.

4 Experiment

4.1 Experimental Settings

All the experiments were run on multiple high-performance servers, each of which is equipped with Intel(R) Xeon(R) Gold 6248 CPU, NVIDIA GV100GL GPU, and 106GB RAM. All deep learning models and proposed corruptions are implemented with Python 3.7, Anaconda 4.5.11, CUDA 10.1 and PyTorch 1.6.0. We spent 1680 h training the selected object detectors and 1800 h on generating COCO-C and further evaluation, respectively.

4.2 RQ1: Potential risks posted by real-world corruptions

Table 2 shows the experimental results of RQ1, where we evaluate the CE of the selected twelve object detectors' on COCO-C. To ensure a fair comparison, we re-train all these object detectors on the original training data of COCO and summarized their performance on the clean test set of COCO in Table 3. As we can see from Table 3, the mean Average Precision (mAP) of each object detectors are consistent with their reported value in original publications, showing that our re-implementation and re-training do not affect the performance of these object detector. The goal of COCO-C is to evaluate the general performance of object

[3] According to Liu et al. (2020) the bounding box is considered correct only if the error rate $err^d_{O_0} < 0.5$.

Table 2 Corruption error (%) on COCO-C of different object detectors with standard training

Detectors (%)	Clean	Camera damage					Image signal processor damage								Board-level damage		
		F	LO	FM-D	CCD-D	CMOS-D	I-BLC	E-BLC	LSC-D	AWB-D	BPC-D	CFAI-D	GC-d	CSC-D	SE	ME	THE
Two-stage																	
FRC	46.6	63.1	55.7	59.9	59.2	67.6	52.4	63.3	52.9	48.8	63.0	50.6	53.8	82.1	49.1	47.0	69.8
MRC	44.8	56.1	52.8	57.6	54.1	65.5	47.5	51.1	50.1	45.9	63.2	48.0	50.6	76.2	47.0	45.0	67.4
RN	43.6	55.9	52.5	56.5	54.0	63.9	47.5	52.7	49.3	45.1	60.3	46.9	49.9	79.6	45.8	46.3	66.5
CRC	41.5	51.4	49.6	54.9	50.5	62.2	44.3	47.8	46.8	42.7	60.0	44.6	47.6	74.4	43.5	41.5	64.0
CMRC	46.2	54.6	54.8	58.4	56.6	67.4	48.7	51.3	51.2	47.5	65.3	49.4	51.7	75.8	51.5	46.5	69.9
HTC	45.4	59.2	56.2	61.9	58.5	70.7	51.0	55.3	54.1	49.0	68.4	50.3	54.5	78.1	50.7	47.4	69.7
One-stage																	
SSD	57.9	68.6	67.0	69.8	69.6	74.7	60.6	61.6	61.5	59.0	71.0	60.2	62.7	84.7	59.8	58.2	78.6
YL3	60.9	70.7	67.0	71.3	70.4	80.6	63.5	63.7	64.0	62.1	73.6	63.3	64.6	82.8	62.8	61.0	81.0
YLA	60.3	68.5	67.0	69.4	69.8	77.7	62.7	63.4	63.9	61.5	73.5	62.5	64.7	84.5	62.1	60.5	80.5
CN	53.6	61.3	63.2	64.1	66.6	76.0	55.5	55.9	56.4	54.8	72.4	55.9	58.8	82.5	55.4	53.8	80.0
Trans-former																	
DETR	43.7	53.5	52.8	53.7	53.5	65.5	46.4	49.8	47.5	45.0	66.0	46.7	49.2	80.7	45.4	43.9	70.3
DDETR	42.9	52.1	59.2	58.9	58.4	67.5	46.5	49.3	51.0	44.3	73.7	46.7	48.4	78.2	46.1	43.4	69.7
Average	48.9	59.6	58.2	61.4	60.1	69.9	52.2	55.4	54.1	50.5	67.5	52.1	54.7	80.0	51.6	49.5	72.3
Quality	SSIM	0.82	0.92	0.98	0.93	0.91	0.94	0.94	0.36	0.99	0.44	0.98	0.82	0.74	0.96	0.99	0.93
	IFC	1.49	1.15	0.82	1.09	0.70	1.89	1.89	0.41	2.03	0.23	1.77	1.52	0.30	2.02	2.18	0.65
	VIF	0.467	0.41	0.29	0.40	0.27	0.62	0.62	0.15	0.72	0.10	0.65	0.55	0.12	0.69	0.74	0.26

A higher error rate (%) means worse performance

Table 3 mAP of detectors implemented by official and our benchmark

Detectors	Official	Benchmark
Two-stage		
FRC	21.2	41.6
MRC	39.8	42.7
RN	40.8	40.8
CRC	42.8	42.5
CMRC	45.8	45.6
HTC	47.1	47.0
One-stage		
SD	33.0	33.7
YL3	28.8	29.5
YLA	28.2	30.5
CN	30.0	29.5
Transformer		
DETR	42.0	40.1
DDETR	46.2	46.8

A higher mAP means better performance

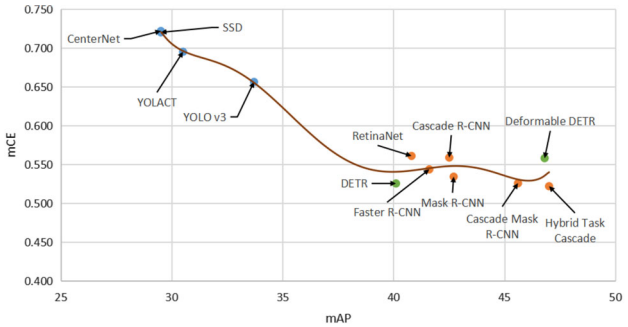

Fig. 8 mCE of different object detectors (orange, blue, and green circles represent two-stage, one-stage, and transformer-based architectures, respectively). Detectors with high mAP and low mCE are preferred

detectors in various real-world scenarios. COCO-C is not designed for training-time optimizations but augmentations with other/similar corruptions are allowed and should be explicitly stated.

According to the results in Table 2, we can find the difference between object detectors on clean data and corrupted data are significant (t-test $= 3.02$, $p < 0.0001$), where there is an average 10% increase on CE across different object detectors and corruption patterns.

Despite the impressive results on clean inputs (*i.e.*, COCO), the current object detectors cannot deliver good performance on real-world corrupted inputs (*i.e.*, COCO-C). From Table 2, among all corruptions, a minimum of 49.8% CE is derived (increased by 1.2% compared to CE_{clean}). Hence our proposed real-world corruptions could fulfill the goal as a performance benchmark for future studies on improving object detector's robustness. Besides, we also find CE_{CMOS-D}, CE_{BPC-D}, CE_{CSC-D}, and CE_{THE} are abnormally high for all architectures, with values higher than 70%. Further investigation shows that these corruption patterns usually have significant impacts on the colour of the images, not on the illumination as LO, LSC-D and GC-D do.

We summarize the findings from RQ1 as the followings:

- **Finding 1.1. All of the proposed real-world corruptions pose a threat to object detectors.** The selected object detectors have poor performance regardless of the corruption categories, suggesting a general vulnerability of detectors. Meanwhile, the SSIM results for corruptions are almost > 0.8, indicating that the structure in the original data is not lost in the generated data. In addition, the results of IFC and VIF are > 0, which indicates that our corruptions generate realistic data and do not cause severe distortion.
- **Finding 1.2. The object detectors are more sensitive to colour changes.** We find that the selected object detectors are extremely sensitive to CMOS-D, BPC-D, CSC-D, and THE, suggesting that when designing a robust object detector against real-world corruptions, colour-change issues should be carefully addressed.

4.3 RQ2: Robustness of different object detectors

Figure 8 shows the robustness performance of the selected object detectors in this study in terms of the relationship between mAP and mCE. As we can observe from the fitted dashed line Fig. 8, the increase of mAP may be accompanied by the decrease of mCE, especially for object detectors with one-stage architectures. However, when it comes to architectures of two-stage and transformers, the correlation is not applicable. This finding produces an insight that a detector with a high mAP may happen to be vulnerable. Besides, we can also interpret that one-stage methods are more vulnerable to real-world corruptions, which shows much higher mCE compared with two-stage and transformer-based methods.

From the perspective of the object detection architecture, we obtain several interesting findings and summarize as follows:

- **Finding 2.1. The relationship between mAP and mCE of the object detectors does not always follow a negative correlation.** Widely-used metric mAP might not be a good indicator for evaluating object detector's robustness against real-world corruption patterns, especially for the two-stage and transformer-based methods, which has shown good performance on clean dataset in terms of mAP.

Table 4 Changes on mCE of different object detectors under different scenarios

Detectors (%)	Universal detector		Autonomous driving detector	
	Universal robustness	Autonomous driving robustness	Universal robustness	Autonomous driving robustness
Two-stage				
FRC	11.3	6.2	12.8	7.7
MRC	10.9	6.3	9.2	7.5
RN	12.3	6.7	9.6	8.9
CRC	11.5	6.3	8.7	9.3
CMRC	11.6	6.5	8.6	9.1
HTC	11.2	6.2	14.9	6.9
One-stage				
SSD	10.2	1.3	7.5	5.6
YL3	8.0	3.7	8.1	6.6
YLA	8.7	4.0	7.2	7.0
CN	8.8	6.4	10.6	6.6
Transformer				
DETR	11.3	6.6	10.1	9.5
DDETR	12.9	7.3	–	–

The Lower the changes on mCE, the higher the stability on cross-scenario applications

- **Finding 2.2. The cascade architecture and the deformable attention module may not be necessary for robustness.** According to Fig. 8, the detectors with cascade architecture and deformable attention module perform better on mAP, while failing to bring effective improvement on mCE, e.g., DDETR to DETR, CRC to FRC, CMRC to MRC. This implies that the ability of the cascade architecture and the deformable attention module is limited in terms of robustness. The object detectors with these components might be more desirable in scenarios with controlled conditions, where there would be more specialized maintenance regulations for the image pipeline of the object detectors. For example, in medical image analysis, the medical image pipelines are carefully controlled and protected from real-world corruptions. However, in more general usage scenarios, e.g., autonomous driving, object detectors and their image pipelines are exposed to lots of real-world corruptions, and pose higher requirements for robustness.
- **Finding 2.3. The one-stage detectors are more prone to flaws, compared with other damages.** One-stage detectors obtain higher CE and mCE compared with the two-stage and transformer-based detectors. Although they are proposed to reduce computational costs, they sacrifice effectiveness to efficiency. This finding may inspire developers to pay more attention to the flaws in one-stage detectors, since efficiency is not the only thing important for object detection.

4.4 RQ3: Robustness under Different Scenarios

In this RQ, we employ a cross-scenario evaluation to investigate the robustness of object detectors. In specific, we evaluate the mCE of both universal detectors (*i.e.*, object detectors trained on COCO) and autonomous driving detectors (*i.e.*, object detectors trained on BDD100K) over data from multiple scenarios (*i.e.*, COCO-C from universal scenario and BDD100K from autonomous driving scenario). Table 4 shows the changes on mCE under different scenarios. Almost all of the CEs experience rises of various degrees on them, with one exception that DDETR crashes when handling autonomous driving scenarios. From the perspective of the cross-scenario evaluation, we summarize our findings as the followings:

- **Finding 3.1. In the autonomous driving scenario, the robustness of both the universal detectors and the autonomous driving detectors is better than that in the universal scenario.** According to Table 4, all detectors only yield a drop in mCE of less than 10 in the autonomous driving scenario. However, in the universal scenario, most of the detectors yield a larger drop in mCE (the largest detector experiences a 14.9% drop). It shows that object detectors are more than capable of handling autonomous driving scenarios.

- **Finding 3.2. Compared with the two-stage and transformer detectors, the one-stage detectors perform well in terms of stability when coping with the cross-scenario application.** In Table 4, the one-stage detectors yielded smaller mCE drops in all scenarios than the other detectors, especially in cross-scenarios. When developing the object detectors, the unclear application scenarios, *i.e.*, the training scenario and application scenario of the detector does not match, may result in the cross-scenario application. According to Table 4, the one-stage detectors are recommended in this case.
- **Finding 3.3. The object detectors with the multi-scale detection mechanism perform better when dealing with autonomous driving scenarios compared with universal scenarios.** According to Table 4, the performance of different one-stage detectors also varies significantly in handling cross-scenario applications. SSD and YL3 detectors perform much better in autonomous driving scenarios (SSD achieves a maximum improvement of 9% in mCE and YL3 achieves 4.3%), while YLA and CN detectors do not. We further investigate the network architectures of the four detectors and find that the multi-scale detection mechanism of SSD and YL3 might contribute to their performance. With the multi-scale detection mechanism, SSD and YL3 detectors could detect multi-scale objects in autonomous driving scenarios, even those with long distances. This is because in autonomous driving scenarios, the size and distance of objects such as vehicles, pedestrians, and bicycles usually differ significantly, leading to different scales. The multi-scale detection mechanism can effectively cope with these objects of different scales, thus improving the robustness of the object detection.

4.5 RQ4: Effectiveness of robustness enhancement methods

In this RQ, we verify if the widely-used robustness enhancement methods are effective against proposed real-world corruption patterns. Table 5 presents the changes of CE compared to Table 2 after employing one of CLAHE, ST, NLM, and AT, respectively. In general, we find that ST and AT have few effects on enhancing object detection robustness against real-world corruption patterns, while CLAHE and NLM had some effects on certain corruption patterns, but overall the effects are limited. We summarize our findings as the followings:

- **Finding 4.1. When object detection methods suffer from real-world corruptions, the effectiveness of robustness enhancement methods is limited.** By observing the changes in CE listed in Table 5, we find that the majority of CEs show a significant increase. This implies that when a robustness enhancement strategy is employed, it does not actually enhance the robustness of object detectors against real-world corruptions.
- **Finding 4.2. CLAHE could enhance the object detectors against corruptions related to illumination changes.** Our results in Table 5 show that, despite the increase in CE against the other corruptions, there is a decrease in CE when it comes to some certain corruptions, *e.g.*, CE_F and CE_{THE}. While both F and THE are corruptions involving illumination changes, wich could lead to the low contrast of the images. However, with CLAHE, the contrast of images could be increased. Thus, CLAHE could be a promising method for enhancing object detection robustness against illumination corruptions.
- **Finding 4.3. NLM could enhance the object detectors against corruptions involving spotty patterns.** In Table 5, NLM brings about a significant decrease in CE_{BPC-D} of all the detectors. In addition, with NLM, there is only a slight increase in CE_{THE} (maximum 1.8, minimum 0.6) of the detectors. According to Fig. 5, the two corruptions generate data with spotty patterns. Considering NLM would reduce the noises in the images, the spotty patterns posed by the corruptions could be reduced by NLM, making the object detectors more robust.
- **Finding 4.4. ST and AT generate unrealistic data, making it ineffective to enhance object detection robustness against real-world corruptions.** As we can see from Table 5, ST and AT cannot improve object detection robustness. Instead, it might have a dramatic increase in CE. We further study the data generated by ST and AT and find that it is severely distorted. Such data, which is less likely to occur in the real-world, would have negative impacts on object detection performance.

4.6 RQ5: Types of flaws revealed in object detectors

From the above RQs, we can find that flaws exist in object detectors in general, even if they are enhanced. This inspires us to further investigate the flaw symptoms in object detectors. To this end, we categorized the detected flaws in RQ2-RQ4 into five different categories of typical flaw symptoms, namely CBWO, WBCO, WBWO, PMD, and MD, as introduced in Sect. 3.5. The results presented in Fig. 9 show the frequency of different flaw symptoms revealed in the twelve selected object detectors. From the perspective of the flaw symptoms, we obtain several interesting findings:

- **Finding 5.1. If a detector finds a correct bounding box of an object, there is a large probability that it would categorized the object into the correct cate-**

Table 5 Changes on CE of different object detectors after applying enhancement strategies. A positive change means the performance is improved

Detectors (%)	Camera damage					Image signal processor damage								Board-level damage		
	F	LO	FM-D	CCD-D	CMOS-D	I-BLC	E-BLC	LSC-D	AWB-D	BPC-D	CFAI-D	GC-d	CSC-D	SE	ME	THE
Two-stage																
CLAHE	**−0.8**	1.7	1.5	2.9	0.5	0.5	0.3	1.3	3.0	**−0.1**	4.4	**−0.2**	**−4.2**	2.2	8.0	0.6
ST	32.8	36.6	33.1	36.9	29.4	36.1	30.8	35.3	38.4	29.5	38.5	35.1	19.2	39.1	39.1	17.1
NL-means	27.5	8.3	10.4	7.4	7.1	10.2	13.6	9.4	6.7	**−2.0**	7.9	5.2	3.8	6.4	6.0	1.8
AT	22.3	12.4	13.1	17.9	15.3	17.0	19.1	13.8	16.2	11.5	15.9	15.0	14.8	15.6	16.3	12.8
One- stage																
CLAHE	**−2.3**	1.4	1.9	0.4	2.2	**−0.7**	**−0.8**	0.4	0.7	3.3	1.4	**−0.7**	1.0	1.1	1.1	1.0
ST	25.2	26.6	24.7	26.1	19.3	27.8	27.2	27.9	28.8	22.1	29.2	25.8	13.7	29.0	26.7	12.0
NL-means	19.3	5.4	6.1	4.9	5.6	6.2	9.1	5.8	3.1	**−2.4**	3.6	3.5	**−0.6**	3.9	3.7	0.6
AT	13.0	9.4	8.3	9.2	6.7	11.7	10.9	10.3	11.2	7.2	10.9	11.5	6.7	10.9	11.5	5.8
Trans- former																
CLAHE	**−1.7**	4.1	3.4	3.6	3.5	1.0	**−0.9**	1.4	3.1	4.2	3.7	**−0.1**	7.3	3.6	3.5	0.1
ST	40.8	39.3	38.7	39.9	31.8	43.6	40.9	42.1	46.0	26.5	45.2	42.1	22.2	46.7	46.8	28.0
NL-means	29.8	9.5	11.3	9.1	9.9	10.2	13.6	9.9	6.7	**−4.7**	6.9	5.1	3.6	6.9	6.6	1.2
AT	17.0	10.3	8.7	11.9	8.6	11.0	12.6	9.2	10.9	7.1	10.3	9.8	7.1	10.7	10.8	8.5

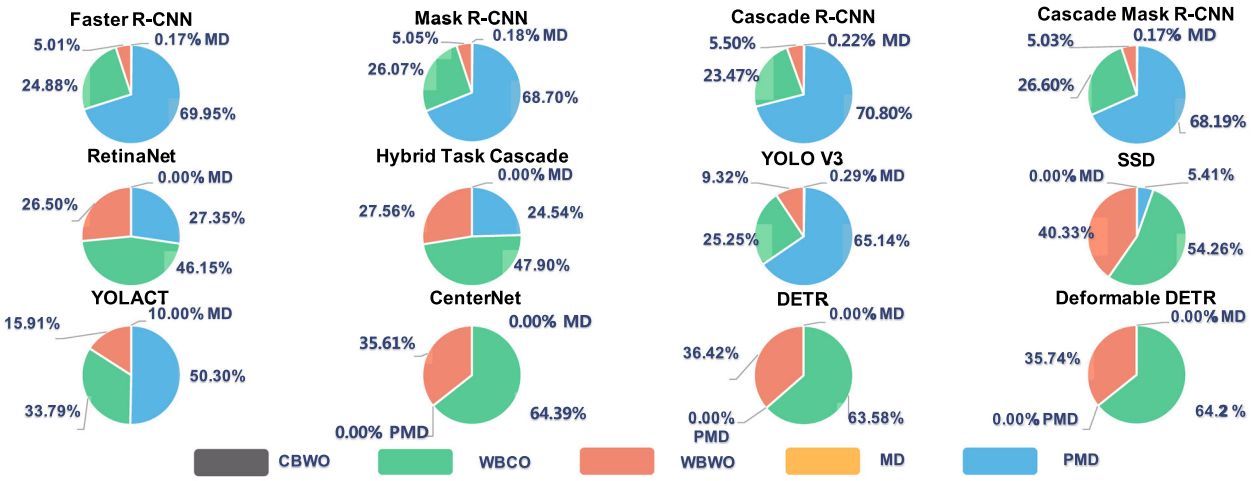

Fig. 9 Distribution of flaws revealed in different object detectors

gory. Intuitively, for two sub-tasks in object detection, *i.e.*, object localization and classification, both of them might be wrong. However, the results in Fig. 9 show that when the bounding box is located correctly, the classification barely goes wrong. This finding poses a challenge to bounding box localization.

- **Finding 5.2. There is a high probability that the detectors could recognize the correct objects even if they do not locate the correct bounding box.** According to Fig. 9, when the detectors do not miss the objects, the most frequent flaws are WBCO flaws. It shows that the detector recognizes the object instances correctly even though the predicted bounding box is incorrect. The performance drop on predicting bounding boxes does not affect the detector's classification of objects' categories.
- **Finding 5.3. Two-stage and one-stage object detectors suffer from the missed detection more compared with transformer-based methods.** As shown in Fig. 9, except for CenterNet, most of the models with two-stage or one-stage architectures exhibit a large number of PMD flaws. However, no PMD flaws are revealed in the two transformer-based methods. Different from other flaw symptoms, (P)MD might lead to more severe outcomes in real-world (*e.g.*, the missed detection of cars in autonomous driving systems may cause car crashes), which poses an urgent need to repair one-stage and two-stage methods to avoid (P)MD.
- **Finding 5.4. Anchor-free mechanisms of object detectors could prevent (P)MD flaws.** According to Fig. 9, the CE detector does not suffer from (P)MD flaws, which is the same as DETR and DDETR detectors. The CE detector employs a center-keypoint detection mechanism to locate objects without relying on anchors. Similarly, the transformer-based object detectors, DETR and DDETR, could locate objects without anchors via the attention mechanism. With these anchor-free mechanisms, the detectors could adjust the localization strategy based on the actual size, shape and location of objects, to ensure adaptive localization and reduce the (P)MD flaws.

5 Threats to validity

In terms of **internal validity**, one potential threat is that the behavior of an object detector can vary when using different environment parameters. To mitigate this threat, we chose to use the same parameters as described in the official documentation of each detector to keep consistency. Further, we confirmed that our results, *i.e.*, the performance of detectors are consistent with their source descriptions and demos (see Table 3). In terms of **external validity**, one potential threat is that our analysis results may not be generalized to other object detectors. To mitigate this threat, we tried our best to collect diverse categories of object detection architectures with SOTA performance. In terms of **construct validity**, one potential threat is that the evaluation metrics may not fully describe the performance of object detectors. To mitigate this threat, we use two different metrics and five different severity levels of corruptions to comprehensively analyze the performance and robustness of the object detectors in our benchmark.

6 Conclusion

In this paper, we present a public benchmark for evaluating object detection robustness. To the best of our knowledge, this is the first benchmark on object detectors in terms of

their robustness against real-world image corruptions. To this end, we propose 16 real-world image corruptions based on the potential damages in real-world image pipeline. Then we leverage two large-scale object detection datasets, *i.e.*, COCO and BDD100K, to create an 80 times larger one based on the proposed image corruptions, namely COCO-C and BDD100K-C. We evaluate 12 representative object detectors covering three different model architectures (*i.e.*, two-stage, one-stage, and transformer) on COCO-C, where our evaluation results and findings show that different kinds of flaws existed in these object detectors, posing an urgent need in the community on designing robust object detectors. Furthermore, our analysis of two widely-used robustness enhancement techniques motivates further improvement on enhancing object detection robustness, in order to build safe and reliable object detection methods for safety-critical applications.

Acknowledgements The authors would like to thank the anonymous reviewers for their insightful comments. This work is supported partially by the National Natural Science Foundation of China (61932012, 62141215, 62372228), Science, Technology, and Innovation Commission of Shenzhen Municipality (CJGJZD20200617103001003), Canada CIFAR AI Chairs Program, the Natural Sciences and Engineering Research Council of Canada (NSERC No.RGPIN-2021-02549, No.RGPAS-2021-00034, No.DGECR-2021-00019), as well as JST-Mirai Program Grant No.JPMJMI20B8, JSPS KAKENHI Grant No.JP21H04877, No.JP23H03372, JP24K02920, and also with the support from TIER IV, Inc. and Autoware Foundation. Chunrong Fang, Jia Liu and Zhenyu Chen are the corresponding authors.

References

Antilogus, P., Astier, P., Doherty, P., Guyonnet, A., & Regnault, N. (2014). The brighter-fatter effect and pixel correlations in ccd sensors. *J. Instrum., 9*(03), C03048.

Bolya, D., Zhou, C., Xiao, F., & Lee, Y. J. (2019). YOLACT: Real-time instance segmentation. In: 2019 IEEE/CVF international conference on computer vision, ICCV 2019, Seoul, Korea (South), October 27–November 2, 2019 (pp. 9156–9165). IEEE. https://doi.org/10.1109/ICCV.2019.00925

Bruneton, E., & Neyret, F. (2008). Precomputed atmospheric scattering. In *Computer graphics forum, Wiley Online Library*, (Vol. 27, pp. 1079–1086).

Buades, A., Coll, B., & Morel, J. (2005). A non-local algorithm for image denoising. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA* (pp. 60–65). IEEE Computer Society. https://doi.org/10.1109/CVPR.2005.38

Cai, Z., & Vasconcelos, N. (2019). Cascade R-CNN: High quality object detection and instance segmentation. *CoRR*, arXiv:1906.09756

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. In Vedaldi, A., Bischof, H., Brox, T., Frahm, J. (Eds.), *Computer vision - ECCV 2020 - 16th European conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I, Springer, Lecture Notes in Computer Science* (Vol. 12346, pp. 213–229). https://doi.org/10.1007/978-3-030-58452-8_13

Celestre, R., Rosenberger, M., & Notni, G. (2016). A novel algorithm for bad pixel detection and correction to improve quality and stability of geometric measurements. *Journal of Physics: Conference Series, 772*, 012002.

Chandra, M., Agarwal, D., & Bansal, A. (2016). Image transmission through wireless channel: A review. In *2016 IEEE 1st international conference on power electronics, intelligent control and energy systems (ICPEICES)* (pp. 1–4). IEEE.

Chaves-González, J. M., Vega-Rodríguez, M. A., Gómez-Pulido, J. A., & Sánchez-Pérez, J. M. (2010). Detecting skin in face recognition systems: A colour spaces study. *Digital Signal Processing, 20*(3), 806–823.

Chen, K., Pang, J., Wang, J., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Shi, J., Ouyang, W., Loy, C. C., & Lin, D. (2019a). Hybrid task cascade for instance segmentation. In *IEEE conference on computer vision and pattern recognition, CVPR 2019, Long Beach, CA, USA, June 16–20, 2019* (pp. 4974–4983). Computer Vision Foundation/IEEE. https://doi.org/10.1109/CVPR.2019.00511, arXiv:1901.07518

Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., et al. (2019b). Mmdetection: Open mmlab detection toolbox and benchmark. arXiv preprint arXiv:1906.07155

CNN. (2016). Who's responsible when an autonomous car crashes? http://money.cnn.com/2016/07/07/technology/tesla-liability-risk/index.html

Dong, Y., Fu, Q., Yang, X., Pang, T., Su, H., Xiao, Z., & Zhu, J. (2020). Benchmarking adversarial robustness on image classification. In *2020 IEEE/CVF conference on computer vision and pattern recognition, CVPR 2020, Seattle, WA, USA, June 13–19, 2020* (pp 318–328). Computer Vision Foundation/IEEE. https://doi.org/10.1109/CVPR42600.2020.00040, https://ieeexplore.ieee.org/document/9157625

Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., & Tian, Q. (2019). Centernet: Keypoint triplets for object detection. In *2019 IEEE/CVF international conference on computer vision, ICCV 2019, Seoul, Korea (South), October 27–November 2, 2019* (pp 6568–6577). IEEE. https://doi.org/10.1109/ICCV.2019.00667

Elharrouss, O., Almaadeed, N., & Al-Máadeed, S. (2021). A review of video surveillance systems. *The Journal of Visual Communication and Image Representation, 77*, 103116. https://doi.org/10.1016/j.jvcir.2021.103116

Erhan, D., Szegedy, C., Toshev, A., & Anguelov, D. (2014a). Scalable object detection using deep neural networks. In *2014 IEEE conference on computer vision and pattern recognition, CVPR 2014, Columbus, OH, USA, June 23–28, 2014* (pp 2155–2162). IEEE Computer Society. https://doi.org/10.1109/CVPR.2014.276

Erhan, D., Szegedy, C., Toshev, A., & Anguelov, D. (2014b). Scalable object detection using deep neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23–28, 2014* (pp 2155–2162). IEEE Computer Society. https://doi.org/10.1109/CVPR.2014.276

Feng, D., Haase-Schütz, C., Rosenbaum, L., Hertlein, H., Gläser, C., Timm, F., Wiesbeck, W., & Dietmayer, K. (2021). Deep multimodal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *The IEEE Transactions on Intelligent Transportation Systems, 22*(3), 1341–1360. https://doi.org/10.1109/TITS.2020.2972974

Fischler, M. A., & Elschlager, R. A. (1973). The representation and matching of pictorial structures. *IEEE Trans Computers, 22*(1), 67–92. https://doi.org/10.1109/T-C.1973.223602

Fossum, E. R. (1997). Cmos image sensors: Electronic camera-on-a-chip. *IEEE Transactions on Electron Devices, 44*(10), 1689–1698. https://doi.org/10.1109/16.628824

Garcia, J., Feng, Y., Shen, J., Almanee, S., Xia, Y., & Chen, Q. A. (2020). A comprehensive study of autonomous vehicle bugs. In Rothermel, G., & Bae, D. (Eds.), *ICSE'20: 42nd international conference on software engineering, Seoul, South Korea, 27 June–19 July, 2020* (pp. 385–396). ACM. https://doi.org/10.1145/3377811.3380397

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F.A., & Brendel, W. (2019). Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *7th international conference on learning representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, OpenReview.net, https://openreview.net/forum?id=Bygh9j09KX

Guo, Q., Strauss, K., Ceze, L., & Malvar, H. S. (2016). High-density image storage using approximate memory cells. In *Proceedings of the twenty-first international conference on architectural support for programming languages and operating systems, Association for Computing Machinery, New York, NY, USA, ASPLOS'16* (pp. 413–426). https://doi.org/10.1145/2872362.2872413

He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2017). Mask r-cnn. In *2017 IEEE international conference on computer vision (ICCV)*.

Hendrycks, D., & Dietterich, T. G. (2019). Benchmarking neural network robustness to common corruptions and perturbations. In *7th international conference on learning representations, ICLR 2019, New Orleans, LA, USA, May 6–9, 2019*, OpenReview.net, https://openreview.net/forum?id=HJz6tiCqYm

Islam, M.J., Nguyen, G., Pan, R., & Rajan, H. (2019). A comprehensive study on deep learning bug characteristics. In Dumas, M., Pfahl, D., Apel, S., & Russo, A. (Eds.), *Proceedings of the ACM joint meeting on European software engineering conference and symposium on the foundations of software engineering, ESEC/SIGSOFT FSE 2019, Tallinn, Estonia, August 26–30, 2019* (pp. 510–520). ACM. https://doi.org/10.1145/3338906.3338955

Kamann, C., & Rother, C. (2021). Benchmarking the robustness of semantic segmentation models with respect to common corruptions. *International Journal of Computer Vision, 129*(2), 462–483. https://doi.org/10.1007/s11263-020-01383-2

Kawamura, S. (1998). Capturing images with digital still cameras. *IEEE Micro, 18*(6), 14–19. https://doi.org/10.1109/40.743680

Kim, K., Kim, J., Song, S., Choi, J. H., Joo, C., & Lee, J. S. (2021). Light lies: Optical adversarial attack. arXiv preprint arXiv:2106.09908

Lin, H. Y., Gu, K. D., & Chang, C. H. (2012). Photo-consistent synthesis of motion blur and depth-of-field effects with a real camera model. *Image and Vision Computing, 30*(9), 605–618.

Lin, T., Goyal, P., Girshick, R. B., He, K., & Dollár, P. (2017). Focal loss for dense object detection. *CoRR*, arXiv:1708.02002

Lin, T., Maire, M., Belongie, S.J., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C.L. (2014). Microsoft COCO: Common objects in context. In Fleet, D. J., Pajdla, T., Schiele, B., & Tuytelaars, T. (Eds.), *Computer Vision—ECCV 2014—13th European conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V, Springer, Lecture Notes in Computer Science* (Vol. 8693, pp. 740–755). https://doi.org/10.1007/978-3-319-10602-1_48

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A. W. M., van Ginneken, B., & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical Image Analysis, 42*, 60–88. https://doi.org/10.1016/j.media.2017.07.005

Liu, J., Wu, C., Wang, Y., Xu, Q., Zhou, Y., Huang, H., Wang, C., Cai, S., Ding, Y., Fan, H., & Wang, J. (2019a). Learning raw image denoising with bayer pattern unification and bayer preserving augmentation. In *IEEE conference on computer vision and pattern recognition workshops, CVPR workshops 2019, Long Beach, CA, USA, June 16–20, 2019*. Computer Vision Foundation/IEEE (pp. 2070–2077). https://doi.org/10.1109/CVPRW.2019.00259, arXiv:1904.12945

Liu, L., Li, H., & Gruteser, M. (2019b). Edge assisted real-time object detection for mobile augmented reality. In Brewster, S. A., Fitzpatrick, G., Cox, A. L., Kostakos, V. (Eds.), *The 25th annual international conference on mobile computing and networking, MobiCom 2019, Los Cabos, Mexico, October 21–25, 2019* (pp. 25:1–25:16). ACM. https://doi.org/10.1145/3300061.3300116

Liu, L., Ouyang, W., Wang, X., Fieguth, P. W., Chen, J., Liu, X., & Pietikäinen, M. (2020). Deep learning for generic object detection: A survey. *International Journal of Computer Vision, 128*(2), 261–318. https://doi.org/10.1007/s11263-019-01247-4

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S. E., Fu, C., & Berg, A. C. (2016). SSD: Single shot multibox detector. In Leibe, B., Matas, J., Sebe, N., Welling, M. (Eds.), *Computer vision - ECCV 2016 - 14th European conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I, Springer, Lecture Notes in Computer Science* (Vol. 9905, pp. 21–37). https://doi.org/10.1007/978-3-319-46448-0_2

Liu, Y., Ma, Z., Liu, X., Ma, S., & Ren, K. (2022). Privacy-preserving object detection for medical images with faster R-CNN. *IEEE Transactions on Information Forensics and Security, 17*, 69–84. https://doi.org/10.1109/TIFS.2019.2946476

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision, Kerkyra, Corfu, Greece, September 20–25, 1999* (pp. 1150–1157). IEEE Computer Society. https://doi.org/10.1109/ICCV.1999.790410

Michaelis, C., Mitzkus, B., Geirhos, R., Rusak, E., Bringmann, O., Ecker, A. S., Bethge, M., & Brendel, W. (2019). Benchmarking robustness in object detection: Autonomous driving when winter is coming. *CoRR*, arXiv:1907.07484

Minh, T. N., Sinn, M., Lam, H. T., & Wistuba, M. (2018). Automated image data preprocessing with deep reinforcement learning. arXiv preprint arXiv:1806.05886

Pathak, A. R., Pandey, M., & Rautaray, S. (2018). Application of deep learning for object detection. *Procedia Computer Science, 132*, 1706–1717. https://doi.org/10.1016/j.procs.2018.05.144

Pizer, S. M., Amburn, E. P., Austin, J. D., Cromartie, R., Geselowitz, A., Greer, T., ter Haar, Romeny B., Zimmerman, J. B., & Zuiderveld, K. (1987). Adaptive histogram equalization and its variations. *Computer Vision, Graphics, and Image Processing, 39*(3), 355–368. https://doi.org/10.1016/S0734-189X(87)80186-X

Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M. E. P., Shyu, M., Chen, S., & Iyengar, S. S. (2019). A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys, 51*(5), 92:1-92:36. https://doi.org/10.1145/3234150

Rahman, S., Rahman, M. M., Abdullah-Al-Wadud, M., Al-Quaderi, G. D., & Shoyaib, M. (2016). An adaptive gamma correction for image enhancement. *EURASIP Journal on Image and Video Processing, 1*, 1–13.

Rebuffi, S. A., Gowal, S., Calian, D. A., Stimberg, F., Wiles, O., & Mann, T. A. (2021). Data augmentation can improve robustness. *Neural Information Processing Systems, 34*, 29935–29948.

Redmon, J., Divvala, S. K., Girshick, R. B., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *2016 IEEE conference on computer vision and pattern recognition, CVPR 2016, Las Vegas, NV, USA, June 27–30, 2016* (pp. 779–788). IEEE Computer Society. https://doi.org/10.1109/CVPR.2016.91

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *CoRR*, arXiv:1804.02767

Ren, S., He, K., Girshick, R. B., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. *CoRR*, arXiv:1506.01497

Schwartz, E., Giryes, R., & Bronstein, A. M. (2019). Deepisp: Toward learning an end-to-end image processing pipeline. *IEEE Transactions on Image Processing, 28*(2), 912–923. https://doi.org/10.1109/TIP.2018.2872858

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2014). Overfeat: Integrated recognition, localization and detection using convolutional networks. In Bengio, Y., LeCun, Y. (Eds.), *2nd international conference on learning representations, ICLR 2014, Banff, AB, Canada, April 14–16, 2014, Conference Track Proceedings*, arXiv:1312.6229

She, Q., Feng, F., Hao, X., Yang, Q., Lan, C., Lomonaco, V., Shi, X., Wang, Z., Guo, Y., Zhang, Y., Qiao, F., & Chan, R.H.M. (2020). Openloris-object: A robotic vision dataset and benchmark for lifelong deep learning. In *2020 IEEE international conference on robotics and automation, ICRA 2020, Paris, France, May 31–August 31, 2020* (pp. 4767–4773). IEEE. https://doi.org/10.1109/ICRA40945.2020.9196887

Sheikh, H. R., & Bovik, A. C. (2006). Image information and visual quality. *IEEE Transactions on Image Processing, 15*(2), 430–444. https://doi.org/10.1109/TIP.2005.859378

Sheikh, H. R., Bovik, A. C., & de Veciana, G. (2005). An information fidelity criterion for image quality assessment using natural scene statistics. *IEEE Transactions on Image Processing, 14*(12), 2117–2128. https://doi.org/10.1109/TIP.2005.859389

Shekar, A. K., Gou, L., Ren, L., & Wendt, A. (2021). Label-free robustness estimation of object detection cnns for autonomous driving applications. *International Journal of Computer Vision, 129*, 1185–1201.

Shen, D., Wu, G., & Suk, H. I. (2017). Deep learning in medical image analysis. *Annual Review of Biomedical Engineering, 19*, 221–248. https://doi.org/10.1146/annurev-bioeng-071516-044442

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data, 6*(1), 1–48.

Silva, V. D., Chesnokov, V., & Larkin, D. (2016). A novel adaptive shading correction algorithm for camera systems. In *Digital Photography and Mobile Imaging*, https://api.semanticscholar.org/CorpusID:36655918

Sindagi, V. A., & Patel, V. M. (2018). A survey of recent advances in cnn-based single image crowd counting and density estimation. *Pattern Recognition Letters, 107*, 3–16. https://doi.org/10.1016/j.patrec.2017.07.007

Sobh, I., Hamed, A., Kumar, V. R., & Yogamani, S. (2021). Adversarial attacks on multi-task visual perception for autonomous driving. arXiv preprint arXiv:2107.07449

Sun, Y., Wang, X., & Tang, X. (2015). Deeply learned face representations are sparse, selective, and robust. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2892–2900)

Szegedy, C., Toshev, A., & Erhan, D. (2013). Deep neural networks for object detection. In Burges, C. J. C., Bottou L, Ghahramani, Z., & Weinberger, K. Q. (Eds.), *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5–8, 2013, Lake Tahoe, Nevada, United States* (pp. 2553–2561). https://proceedings.neurips.cc/paper/2013/hash/f7cade80b7cc92b991cf4d2806d6bd78-Abstract.html

Tian, Y., Pei, K., Jana, S., & Ray, B. (2018). Deeptest: automated testing of deep-neural-network-driven autonomous cars. In Chaudron, M., Crnkovic, I., Chechik, M., Harman, M. (Eds.), *Proceedings of the 40th international conference on software engineering, ICSE 2018, Gothenburg, Sweden, May 27–June 03, 2018* (pp. 303–314). ACM. https://doi.org/10.1145/3180155.3180220

Times, T. N. Y. (2017). Tesal's self-driving system cleared in deadly crash. https://www.nytimes.com/2017/01/19/business/tesla-model-s-autopilot-fatal-crash.html

Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., & McDaniel, P. (2018). Ensemble adversarial training: Attacks and defenses. In *International conference on learning representations*, https://openreview.net/forum?id=rkZvSe-RZ

Uricar, M., Sistu, G., Rashed, H., Vobecky, A., Kumar, V.R., Krizek, P., Burger, F., & Yogamani, S. (2021). Let's get dirty: Gan based data augmentation for camera lens soiling detection in autonomous driving. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision (WACV)* (pp. 766–775)

Volos, C. K., Kyprianidis, I. M., & Stouboulos, I. N. (2013). Image encryption process based on chaotic synchronization phenomena. *Signal Processing, 93*(5), 1328–1340.

Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing, 13*(4), 600–612. https://doi.org/10.1109/TIP.2003.819861

Wu, B., Iandola, F.N., Jin, P.H., & Keutzer, K. (2017). Squeezedet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving. In *2017 IEEE conference on computer vision and pattern recognition workshops, CVPR workshops 2017, Honolulu, HI, USA, July 21–26, 2017* (pp. 446–454). IEEE Computer Society. https://doi.org/10.1109/CVPRW.2017.60

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

Ying, J., He, Y., & Zhou, Z. (2009). Analysis on laser spot locating precision affected by cmos sensor fill factor in laser warning system. In *2009 9th international conference on electronic measurement & instruments* (pp. 2-202–2-206). https://doi.org/10.1109/ICEMI.2009.5274607

Zhang, Y., Dong, B., & Heide, F. (2022). All you need is raw: Defending against adversarial attacks with camera image pipelines. In *European conference on computer vision* (pp. 323–343). Springer.

Zhong, Z., Zheng, L., Kang, G., Li, S., & Yang, Y. (2020). Random erasing data augmentation. In: *Proceedings of the AAAI conference on artificial intelligence,* (Vol. 34, pp.13001–13008).

Zhou, J., & Glotzbach, J. (2007). Image pipeline tuning for digital cameras. In *2007 IEEE international symposium on consumer electronics* (pp. 1–4). IEEE. https://doi.org/10.1109/ISCE.2007.4382143

Zhu, X., Su, W., Lu, L., Li, B., Wang, X., & Dai, J. (2021). Deformable DETR: Deformable transformers for end-to-end object detection. In *9th international conference on learning representations, ICLR 2021, Virtual Event, Austria, May 3–7, 2021*, OpenReview.net, https://openreview.net/forum?id=gZ9hCDWe6ke