Navigating the Waters of Object Detection: Evaluating the Robustness of Real-time Object Detection Models for Autonomous Surface Vehicles

Yunjia Wang
Department of Computer Science
KU Leuven
Bruges, Belgium
yunjia.wang@kuleuven.be

Kaizheng Wang
Department of Computer Science
KU Leuven
Bruges, Belgium
kaizheng.wang@kuleuven.be

Zihao Zhang
School of Engineering and Applied Science
Columbia University
New York, USA
zz2763@columbia.edu

Jeroen Boydens
Department of Computer Science
KU Leuven
Bruges, Belgium
jeroen.boydens@kuleuven.be

Davy Pissoort
Department of Electrical Engineering
KU Leuven
Bruges, Belgium
davy.pissoort@kuleuven.be

Mathias Verbeke
Department of Computer Science
KU Leuven
Bruges, Belgium
mathias.verbeke@kuleuven.be

***Abstract*—The safe navigation of Autonomous Surface Vehicles (ASVs) critically depends on their ability to detect objects, such as other vessels or obstacles. However, the variable and often harsh environment, characterized by fluctuating weather and image distortions, presents significant challenges for reliable object detection. To ensure a safe system operation, it is imperative to understand the extent of these challenges and identify specific vulnerabilities. In this paper[1], we evaluate the corruption robustness of state-of-the-art real-time object detection models for ASVs. We conduct a comprehensive analysis across various model scales, employing three distinct waterborne object detection datasets. By augmenting each test dataset with 15 types of corruption, we investigate model robustness according to two proposed metrics. Our findings reveal that certain corruption types markedly impair object detection performance, which could pose significant safety risks in autonomous shipping. Conversely, some corruption types have minimal effect on performance, regardless of the model or dataset. Furthermore, the results reveal a notable correlation between the scale of object detection model and its robustness, with larger models generally exhibiting higher resilience to corruption.**

***Index Terms*—corruption robustness, real-time object detection, autonomous surfaces vehicles, model resilience**

I. Introduction

With the rapid development of artificial intelligence and other information and communications technologies which have been widely deployed in autonomous systems, the development of ASVs has increasingly gained attention in the last decade. The concept of an ASV was defined in the European Waterborne Technologies Platform [1], serving as the foundation of autonomous vessels. From then on, many projects on unmanned and autonomous vessels have emerged [2]. Research conducted by the Maritime Unmanned Navigation through Intelligence in Networks (MUNIN) project concluded that, in general, there are no major obstacles to the realisation of fully autonomous vessels, although a few constraints need to be addressed [3]. DNV GL, a company headquartered in Høvik, Norway, has developed an autonomous ship prototype used for Short-Sea-Shipping (SSS), named the ReVolt [4]. The Advanced Autonomous Waterborne Applications Initiative (AAWA) was launched in 2015 aiming to create preliminary designs of autonomous ships [5]. Later, YARA Birkeland, targetting to become the world's first fully electric autonomous and zero emission container feeder, has finished its design relying on the outcome of MUNIN in 2017 and commenced commercial operation with the first set of highly automated systems and onboard crew in 2022.

Generally speaking, ASVs rely on accurate and efficient vision-based object detection as a critical component to navigate safely and make intelligent decisions in real-time. Although sensors such as LiDAR, radar, and sonar are also used to enhance situational awareness, visual cameras are better at object classification and identification, thanks to the wealth of information derived from high-resolution images. Furthermore, object detection is the foundation of many key tasks in ASVs such as object tracking and path planning. There are basically two main categories of deep learning algorithms in object detection: two-phase and single-stage approaches. The two-phase approach first generates a set of region proposals and then classifies each proposal as an object or background. Compared with two-phase approaches, single-stage object detection algorithms do not require an explicit

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 955768 (AUTOBarge) and FET-Open grant agreement No 964505 (E-pi).

[1] For reproducibility, the source code and dataset splits used in the paper are available at https://gitlab.kuleuven.be/m-group-campus-brugge/dtai_public/publications/ieee_cai24.

979-8-3503-5409-6/24/$31.00 ©2024 IEEE
DOI 10.1109/CAI59869.2024.00180

region proposal step, but predict the class and location of objects directly in a single shot, including You Only Look Once (YOLO) [6], Single Shot Multibox Detector (SSD) [7] and NanoDet-Plus [8]. This makes them suitable for real-time applications like ASVs with higher speed and less computational cost, though usually with a trade-off against accuracy [9].

Nevertheless, the adoption of vision-based approaches for ASVs presents a major challenge: ensuring robustness and reliability [10]–[14]. The risks are significant, as object detection failures in ASVs can lead to severe incidents like powered collisions, identified as an unacceptable hazard in maritime operations [10]. Due to the known robustness limitations of vision-based deep learning models, the field has focused on adversarial perturbation robustness and naturally occurring data corruption robustness, the so-called natural robustness or corruption robustness [15], [16]. For natural robustness, benchmarks like IMAGENET-C and IMAGENET-P evaluate image classification models against common corruption and pertubations [15], [17]–[19], and efforts have been made to evaluate object detection in autonomous driving [20]. However, a comprehensive evaluation specifically for ASVs remains conspicuously absent. This gap is critical, considering the existing strategies to enhance model robustness for ASVs, such as data augmentation [21] and network architecture refinement [22], [23], are often limited to few corruption types and one single model, without a collective understanding of the full spectrum of challenges in ASV environments. Our work fills this gap by providing a comprehensive evaluation of corruption robustness for ASVs. The main contributions of this paper are summarized as follows:

- We address a key concern in utilizing vision-based object detection for ASVs by comprehensively evaluating the robustness of leading real-time object detection methods in the context of autonomous shipping. Our analysis spans three waterborne object detection datasets, each with distinct characteristics, providing a broad and insightful assessment of current technological capabilities in this field.
- Our research categorizes the 15 types of corruption we applied into two distinct groups: susceptible and insusceptible corruption. This categorization, based on the preserved model performance, reveals significant gaps in current model robustness against specific corruption types in ASVs. This critical finding not only highlights these gaps but also urgently calls for the development of targeted methodologies to enhance the resilience of these systems, paving the way for safer and more reliable ASV deployment.
- We explore the relatively unexplored territory of the relationship between model scale and corruption robustness in object detection models. Our findings indicate a notable trend: larger models tend to demonstrate enhanced robustness against susceptible corruption. This discovery offers valuable insights into model optimization strategies, particularly in enhancing real-world applicability and performance of ASVs. Our work significantly contributes to the understanding of how model scaling can be used to enhance robustness in practical applications.

II. Methodology

To investigate the efficacy and robustness of real-time object detection approaches in the context of ASVs, our work evaluates the performance of distinct approaches in a spectrum of digital and environmental corruption. Additionally, the influence of model scale on these methodologies' performance is also evaluated and analyzed. In this section, the chosen object detection algorithms, the selection of datasets, the methodology adopted to generate corruption, as well as the evaluation metrics utilized are presented.

A. Real-time object detection approaches

Prioritizing real-time capability as the primary concern while balancing performance in terms of accuracy, the evaluation encompasses three main approaches: YOLOv8 [24], SSD [7], and NanoDet-Plus [8].

1) YOLOv8: YOLOv8 is the latest version of YOLO, launched in January 2023 by Ultralytics. According to Ultralytics, YOLOv8 is a state-of-the-art model that builds on previous YOLO versions with new features. To cater to different scenarios and strike a balance between efficiency and accuracy, YOLOv8 offers five different scales known as n/s/m/l/x. Among these options, YOLOv8n is the smallest model offered, while YOLOv8x is the largest.

In this paper, we use all five models of different scales for robustness evaluation.

2) SSD: SSD is a classic real-time object detection algorithm. SSD forecasts both scores and box offsets using a predefined collection of anchors with varying scales, positioned at each location across multiple feature maps obtained from a Feature Pyramid Network (FPN). SSD can work with different network backbones such as VGG, ResNet, EfficientNet, and MobileNet.

In this paper, we use ResNet as the network backbone of SSD with the implementation in [25]. Specifically, we also consider different sizes of the ResNet backbone, including ResNet-18, ResNet-34 and ResNet-50 [26].

3) NanoDet and NanoDet-Plus: NanoDet is an one-stage anchor-free real-time object detection model which is fast and lightweight. In a recent survey [27], a comprehensive study of various real-time object detection algorithms considering 8 different dimensions including accuracy, speed, natural robustness, and adversarial robustness, is conducted. In the investigation, NanoDet reaches the highest scores on most of the axes. NanoDet-Plus is the next version of NanoDet, which further improves the accuracy with 30% at the cost of an increase of 1ms in latency.

In this work, we use NanoDet-Plus for robustness evaluation with input resolution 416*416. Two different model sizes are provided in [8], NanoDet-Plus-m and NanoDet-Plus-m-1.5x, respectively.

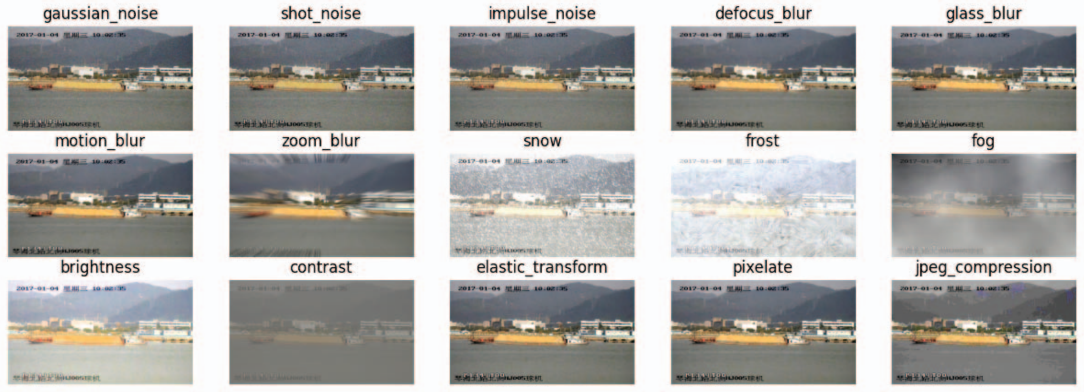


Fig. 1. An example image from the SeaShips dataset under 15 types of corruption at severity 3.

B. Datasets

To enhance dataset diversity, we include datasets with various camera settings (onboard, onshore, and on a drone) and distinct waterborne environments (maritime and inland waterway).

1) SeaShips: Seaships is a large-scale and well-annotated maritime dataset specifically designed for ship object detection [28]. The online released version of SeaShips includes 7000 images in total, consisting of six common ship types: ore carrier, passenger ship, container ship, bulk cargo carrier, general cargo ship, and fishing boat. This dataset is obtained by cameras deployed in different locations onshore. The dataset also includes ship images of various sizes, under different lighting conditions and from different viewpoints.

2) Singapore Maritime Dataset (SMD): SMD is another widely used maritime object detection and object tracking dataset [29]. This dataset provides Visual-Optical (VIS) and Near Infrared (NIR) videos taken from Singapore waters with 10 classes of annotations. In this paper we only consider the VIS videos. The dataset includes VIS videos taken from both the onshore and the onboard cameras. Different illumination conditions such as fog, daylight, and dark/twilight are also covered in this dataset.

3) Shared Situational Awareness between Vessels (SSAVE): SSAVE is a small-scale dataset which contains 827 representative images gathered in various Belgian waterways [30]. Different from SeaShips and SMD, SSAVE is an inland waterway dataset with specific backgrounds and objects such as various types of markers on the water surface. The images in SSAVE are captured either onboard from a navigating barge or from a camera mounted on a drone.

C. Corruption

In existing literature, there are two distinct technical routes for benchmarking corruption robustness: one using corrupted images collected from real-world scenarios and the other using synthesized corruption. Each has its unique strengths and limitations. For synthesized corruption-based benchmarks, creating a test dataset with various severity levels of abundant corrupted images is comparatively easier, more cost-effective, and practical when compared to collecting corrupted images from a wide range of real-world scenarios. Additionally, this approach allows for the reuse of object detection labels, making it a reasonable choice for providing verification of known unknowns. There might be a pertinent concern regarding the extent to which offline evaluations conducted on corruption robustness benchmarks accurately mirror a model's resilience in real-world scenarios. However, improvements in artificial robustness benchmarks are already found to be transferable to real-world distribution shifts in literature [19]. In this work, we evaluate models with synthesized corruption, using 15 different corruption types initially proposed by ImageNet-C [15]. Each corruption includes 5 levels of corruption severity. The 15 corruption types can further be categorized as three categories, related to the lighting condition or the weather condition (Gaussian noise, shot noise, glass blur, snow, frost, fog, brightness, and contrast), related to the camera movement (defocus blur, motion blur, and zoom blur), and related to digital processing or errors that occured during digital processing (impulse noise, elastic transform, pixelation, and JPEG compression), making them also common to ASVs. Fig. 1 depicts an example from the SeaShips dataset corrupted with the 15 corruption types.

D. Metrics

We evaluate the performance of object detection models with two standardized performance metrics: mAP^{50} and mAP, where mAP^{50} denotes the mean Average Precision computed at 50% Intersection over Union (IoU) and mAP denotes the mean Average Precision averaged over 10 IoU values ranging from 0.50 to 0.95 with a step size of 0.05 (.50:.05:.95). The two metrics align with $\mathrm{AP}^{\mathrm{IoU}=.50}$ and AP in the COCO benchmark respectively [31].

As a mathematical definition of robustness is still largely missing in literature, similar to other existing works [20], we consider the robustness here as a model's ability to preserve its model performance under natural corruption. We also assume that a model's ability to preserve its model performance under natural corruption may vary depending on the type of corruption. Based on this assumption, we propose two metrics to evaluate model robustness under a specific corruption type. These metrics are derived from the two standardized performance metrics, mAP^{50} and mAP.

Based on mAP^{50}, we define a robustness metric on a corruption type as the relative mean average precision at 50% IoU on the corruption type c:

$$\mathrm{rmAP}^{50}{}_c = \frac{1}{S}\frac{\sum_{s=1}^{S}\mathrm{mAP}^{50}{}_{s,c}}{\mathrm{mAP}^{50}} \times 100\%, \tag{1}$$

where s denotes the level of severity of this corruption, S is the total number of severity levels (S equals 5 in this work), $\mathrm{mAP}^{50}{}_{s,c}$ stands for the mean Average Precision at IoU 50% obtained under corruption type c at severity level s, and mAP^{50} stands for the measure obtained on the clean test dataset without synthesized corruption. $\mathrm{rmAP}^{50}{}_c$ then is the preserved percentage of mAP^{50} performance under corruption type c averaged across all levels of severity.

Similarly, based on mAP, we define another robustness metric on a corruption type as the relative mean average precision which averages over IoUs between 50% and 95% on corruption type c:

$$\mathrm{rmAP}_c = \frac{1}{S}\frac{\sum_{s=1}^{S}\mathrm{mAP}_{s,c}}{\mathrm{mAP}} \times 100\%, \tag{2}$$

where $\mathrm{mAP}_{s,c}$ stands for the mean Average Precision averaged over IoUs under corruption type c at severity level s, and mAP stands for the measure on the clean test dataset without any synthesized corruption. Similar to $\mathrm{rmAP}^{50}{}_c$, rmAP_c denotes the preserved percentage of mAP performance when subjected to corruption type c, averaged across all levels of severity.

We also define two average robustness metrics to represent the robustness averaged over various corruption types based on $\mathrm{rmAP}^{50}{}_c$ and rmAP_c respectively, which are calculated as follows:

$$\mathrm{rmAP}^{50}_{\mathrm{avg}} = \frac{1}{C}\sum_{c=1}^{C}\mathrm{rmAP}^{50}{}_c, \tag{3}$$

$$\mathrm{rmAP}_{\mathrm{avg}} = \frac{1}{C}\sum_{c=1}^{C}\mathrm{rmAP}_c, \tag{4}$$

where C stands for the total number of corruption types that we focus on. $\mathrm{rmAP}^{50}_{\mathrm{avg}}$ and $\mathrm{rmAP}_{\mathrm{avg}}$ represent a model's average ability to preserve its performance over various defined corruption types.

III. Results and Discussion

In order to assess the performance degradation exhibited by real-time object detection models employed in the context of ASVs when exposed to diverse corruption types, we adopt a structured two-stage methodology to evaluate their relative robustness. The first stage entails an evaluation of YOLOv8, SSD and NanoDet-Plus, all performed on the SeaShips dataset. It is important to note that, for the object detection approaches, varying sizes of models or network backbones are utilized within each approach, as introduced in the previous section. Subsequently, the second stage involves evaluating the same YOLOv8 models on three different datasets: SeaShips (experiments already performed in the first stage), SMD and SSAVE. This two-stage methodology enables us to discover patterns and insights that traverse both varied object detection algorithms and datasets, thereby facilitating a comprehensive exploration of the obtained results.

In each stage, models are evaluated on both the clean and the corrupted test dataset which includes 15 types of corruption at all five severity levels. Every model undergoes a rigorous training and evaluation process that is repeated five times. The results of evaluation obtained from these five runs are subsequently averaged to yield a consolidated performance metric.

Tables I and II show the results of the robustness evaluation experiments in the first stage and the second stage, respectively. Given the space limitation, we present results for only one of the two robustness metrics previously defined, namely $\mathrm{rmAP}^{50}{}_c$. Notably, the other metric, rmAP_c, yields findings highly analogous to those of $\mathrm{rmAP}^{50}{}_c$. The robust linear correlation observed between the two metrics justifies our focus on a single metric for clarity and brevity. The performance of models is variably degraded by different types of corruption. Corruption can notably induce diverse detection failures compared to clean dataset performance, including missed detections (false negatives), particularly prevalent among smaller objects, alongside misclassifications, inaccurate bounding boxes, and spurious detections (false positives). Figure 2 illustrates these failures in a challenging context of overlapping ships. This figure encapsulates all discussed failure types attributable to corruption.

It is somewhat astonishing that, in certain instances, the relative performance for a few corruption types marginally surpasses 100%. This suggests that under such corruption, the model's performance might be slightly better than its performance on the clean dataset. Notably, a majority of these unusual figures originate from the outcomes of the SMD dataset. This observation leads us to posit that this phenomenon could be attributed to the presence of noisy labels and imprecisely located bounding boxes within the dataset, as the labeling issue within the SMD dataset has been acknowledged in other works [32].

A. Susceptible vs. insusceptible corruption

The model robustness across different corruption types is depicted in Fig. 3. This figure, in conjunction with the information presented in Tables I and II, shows that models exhibit varying levels of susceptibility to different types of corruption. Notably, while some corruption types (namely Gaussian noise, shot noise, impulse noise, zoom blur, snow, frost, fog, and contrast) exert a significant impact on these approaches, rendering them highly vulnerable, the other forms of corruption have minimal influence, showcasing a remarkable degree of

TABLE I
rmAP$_c^{50}$ (%) - Robustness of YOLOv8, SSD and NanoDet-Plus on SeaShips dataset

approach	YOLO					SSD			NanoDet-Plus	
model	YOLO v8n	YOLO v8s	YOLO v8m	YOLO v8l	YOLO v8x	SSD-ResNet-18	SSD-ResNet-34	SSD-ResNet-50	NanoDet-Plus-m	NanoDet-Plus-m-1.5x
gaussian_noise	76.1	79.2	84.4	83.8	83.7	58.1	59.4	58.1	56.5	58.9
shot_noise	73.5	75.9	81.2	80.4	80.7	54.4	56.0	55.4	50.6	53.2
impulse_noise	76.2	78.9	84.8	84.1	84.2	57.3	58.4	56.8	55.4	57.9
defocus_blur	98.9	98.5	98.6	98.6	98.6	93.6	94.4	92.0	97.4	97.6
glass_blur	99.8	99.8	99.8	99.8	99.7	97.7	98.4	97.3	99.3	99.2
motion_blur	98.3	99.0	99.2	99.4	99.4	97.2	97.9	96.6	97.9	98.5
zoom_blur	62.3	64.4	63.4	66.2	66.3	57.1	57.8	53.2	77.8	77.1
snow	59.3	64.0	73.4	74.1	74.9	53.7	57.9	61.3	52.4	64.0
frost	54.1	60.4	67.6	68.1	69.1	53.9	59.9	60.8	53.9	54.4
fog	95.0	95.5	97.0	97.7	98.2	81.9	83.6	89.0	72.5	74.0
brightness	99.4	99.4	99.7	99.5	99.5	94.5	95.0	95.6	96.1	96.4
contrast	57.0	59.1	61.7	65.6	69.6	64.0	67.4	74.4	49.2	48.9
elastic_transform	100.0	100.0	100.0	100.0	99.9	99.6	100.3	99.4	100.0	100.0
pixelate	100.0	100.0	100.0	100.0	100.0	99.7	100.3	99.6	100.0	99.9
jpeg_compression	99.7	99.8	99.8	99.9	99.9	99.0	99.5	98.9	99.0	99.4

TABLE II
rmAP$_c^{50}$ (%) - Robustness of YOLOv8 on SMD and SSAVE dataset

dataset	SMD					SSAVE				
model	YOLO v8n	YOLO v8s	YOLO v8m	YOLO v8l	YOLO v8x	YOLO v8n	YOLO v8s	YOLO v8m	YOLO v8l	YOLO v8x
gaussian_noise	56.3	62.6	65.3	64.6	62.9	76.7	79.3	81.4	84.3	84.1
shot_noise	46.2	51.7	55.4	54.9	53.0	78.3	77.8	83.1	84.6	83.1
impulse_noise	55.0	60.9	63.7	63.3	60.5	78.1	80.4	84.0	86.1	84.1
defocus_blur	96.1	96.6	97.6	99.1	97.6	92.5	93.6	95.5	95.1	97.4
glass_blur	98.0	98.7	100.6	101.8	101.2	93.3	93.6	94.6	93.4	95.9
motion_blur	96.6	97.9	98.8	100.0	99.5	87.3	85.5	86.7	85.5	89.1
zoom_blur	42.4	49.2	51.7	52.3	53.7	29.1	33.5	35.7	37.8	40.5
snow	44.8	62.5	67.8	71.0	72.7	75.7	81.7	89.5	88.8	92.4
frost	45.2	61.0	66.4	70.8	75.8	60.7	73.2	83.4	83.8	88.0
fog	90.7	94.8	98.2	99.7	100.5	78.1	87.2	92.8	93.6	98.5
brightness	98.2	99.2	101.1	102.0	101.7	96.5	97.0	98.4	97.7	100.8
contrast	52.9	65.8	69.5	76.1	79.3	45.3	53.7	62.2	62.8	68.8
elastic_transform	100.7	100.6	102.3	102.8	102.4	97.7	98.4	97.7	97.1	100.4
pixelate	101.0	100.8	102.2	103.0	102.1	99.9	100.0	99.1	99.7	102.6
jpeg_compression	96.3	97.0	99.0	100.9	98.1	95.6	96.9	95.9	95.3	98.7

resilience. Specifically, models consistently display analogous susceptibility to identical corruption types, irrespective of the chosen real-time object detection approach or dataset.

Guided by these findings, we group the 15 corruption types into two distinct groups: the susceptible corruption group and the insusceptible corruption group. The susceptible corruption group comprises eight corruption types: Gaussian noise, shot noise, impulse noise, zoom blur, snow, frost, fog, and contrast. Conversely, the insusceptible corruption group consists of seven image corruption types: defocus blur, glass blur, motion blur, brightness, elastic transform, pixelate, and jpeg compression. The visualization of the model robustness in terms of these two corruption groups is shown in Fig. 3.

In the following analysis, our primary focus will be directed towards the susceptible corruption group, given their pronounced impact on real-time object detection models used in ASVs. This emphasis not only illuminates pertinent findings for future research but also holds relevance for industrial applications. The imperative to prioritize and enhance robustness against corruption types falling within the susceptible

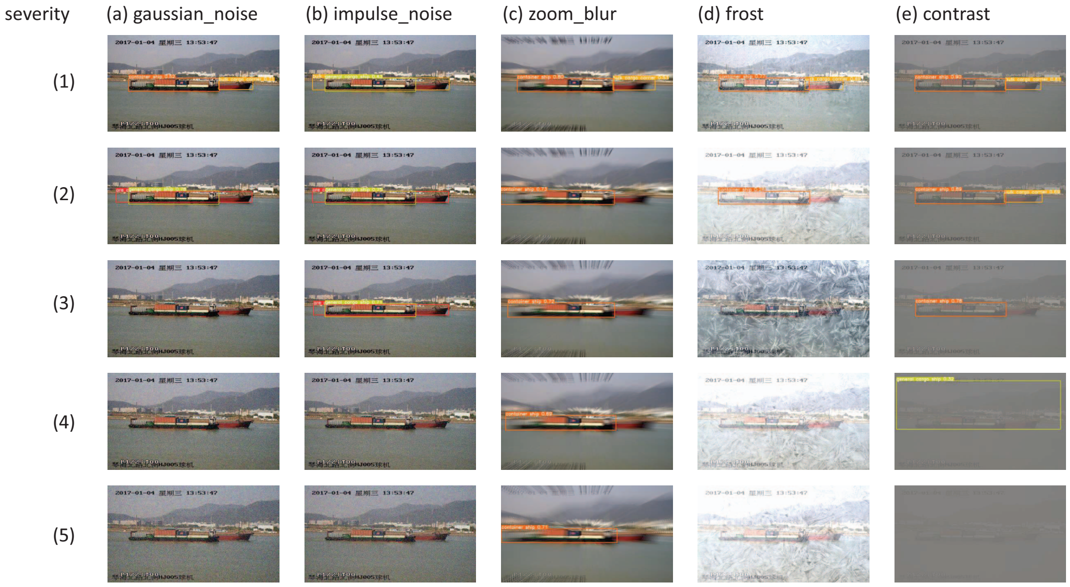
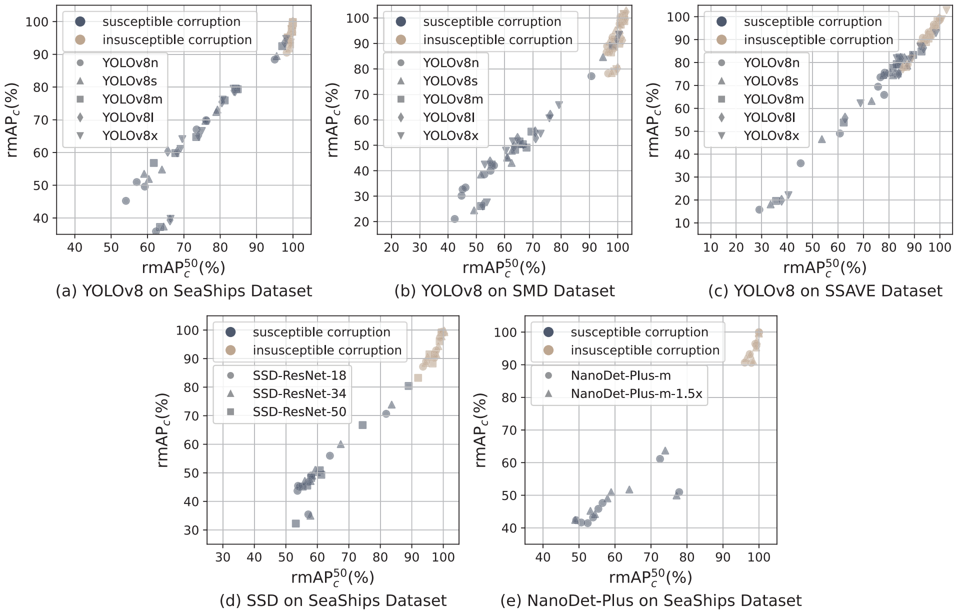


Fig. 2. Detection failures appearing in a challenging scenario with overlapping ships, i.e. a container ship in the front and a bulk cargo carrier at back, under various corruption types. The columns of the figure stand for corruption types while the rows stand for the levels of severity from 1 to 5. Detection failures such as missed detections (e.g. a3), misclassifications (e.g. b1), inaccurate bounding boxes (e.g. e4), and spurious detections (e.g. b3 where the container ship in the front is detected as two ships, a general cargo ship and an ore carrier) are included.



(a) YOLOv8 on SeaShips Dataset

(b) YOLOv8 on SMD Dataset

(c) YOLOv8 on SSAVE Dataset

(d) SSD on SeaShips Dataset

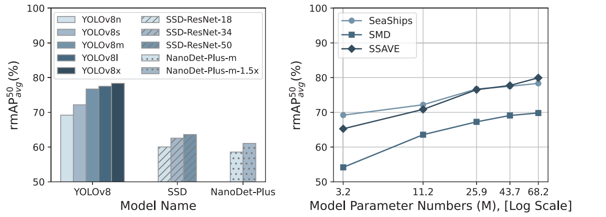
(e) NanoDet-Plus on SeaShips Dataset

Fig. 3. Visualization of the model robustness rmAP^{50}_c and rmAP_c on 15 corruption types. The 15 corruption types are clustered into susceptible corruption and insusceptible corruption, denoted in different colors.

corruption group is thereby underscored.

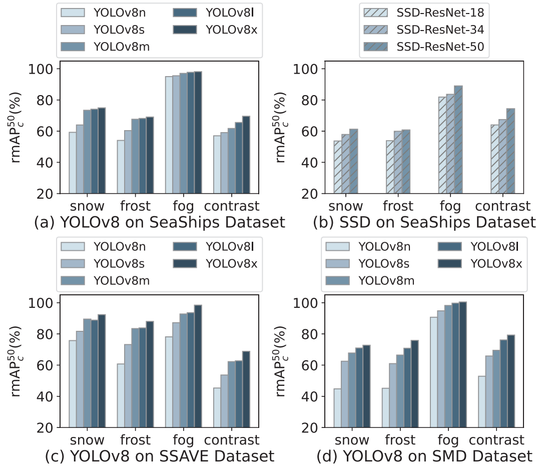
B. Larger models exhibit better robustness

To explore the correlation between model scale and robustness across various corruption types, we conduct a comparative



(a) 3 Approaches of Models on SeaShips (b) $\mathrm{rmAP}^{50}_{avg}(\%)$ - YOLOv8 Models on 3 Datasets

Fig. 4. The average robustness of the susceptible corruption types increases with the model scale. (a) shows the results from the first stage and (b) shows the results from the second stage.



(a) YOLOv8 on SeaShips Dataset (b) SSD on SeaShips Dataset

(c) YOLOv8 on SSAVE Dataset (d) YOLOv8 on SMD Dataset

Fig. 5. The robustness generally increases with model scale.

analysis of average robustness among different model scales. Our focus is directed specifically at susceptibility to corruption types that hold substantial influence over model performance, consequently impacting the safety of ASVs. The results are presented in Fig. 4. From this figure, a prevailing trend becomes apparent: larger models, as adopted in all three examined methodologies, consistently demonstrate higher average robustness across the three different datasets.

Through an in-depth exploration of specific corruption types, we further explore the relationship between the particular susceptible corruption types and model scales, where a similar trend is found. Due to the page limitation, we show the observed trend in Fig. 5 for the SSD and the YOLOv8 approach under the four corruption types directly related to weather conditions: snow, frost, fog, and contrast.

It is crucial to emphasize that the gains in robustness attributed to varying model scales do not uniformly apply to all corruption types. Furthermore, it is worth highlighting that the robustness metrics employed in this study are relative values, reflecting the percentage of model performance preserved under specific corruption types.

IV. Conclusion and Future Work

In this paper, we conducted a comprehensive evaluation of several state-of-the-art real-time object detection approaches, YOLOv8, SSD, and NanoDet-Plus in the context of ASVs. Our evaluation included different model scales on both clean and corrupted datasets generated by adding 15 types of common corruption. We introduced two robustness metrics to analyze the model's performance on corrupted data. Moreover, in the context of autonomous shipping, our exploration uncovers a noteworthy revelation: while state-of-the-art real-time object detection models exhibit susceptibility to specific image corruption types, they demonstrate resilience against others. The results show that existing appoaches for real-time object detection are inadequate in robustness for ASVs under certain types of image corruption, including those related to weather conditions. It suggests the need for methodologies to mitigate this vulnerability in ASVs, such as incorporating strategies for uncertainty monitoring or increasing reliance on alternative sensors under those corruption types.

Notably, we observed a consistent trend, whereby larger models tend to showcase increased robustness against the susceptible corruption types. This trend, evident across multiple datasets and all employed object detection approaches, suggests that model scale plays a key role in shaping corruption robustness. Based on the trend, there is a need to strike a balance for ASVs between the model size and computational capabilities, as model robustness plays a crucial role in ensuring the safe navigation. Our results offer insights into enhancing the robustness of object detection models for real-world applications. However, future research is required to uncover the causes behind the models' lack of robustness. In turn, this can form a solid basis for the development of effective model monitoring strategies, thereby improving the safety and resilience of deep learning models. Furthermore, considering that real-world scenarios often involve multiple types of corruption simultaneously, exploring this aspect represents an intriguing direction for future research.

References

[1] Waterborne Technology Platform, "Waterborne implementation plan: Issue May 2011," 2011.

[2] V. Bertram, "Unmanned & autonomous shipping: A technology review," in *Proceedings of the 10th Symposium on High-Performance Marine Vehicles, Cortona*, 2016, pp. 10–24.

[3] Ø. J. Rødseth, "From concept to reality: Unmanned merchant ship research in norway," *Proceedings of Underwater Technology (UT), IEEE, Busan, Korea*, 2017.

[4] S. Adams, "Revolt–next generation short sea shipping," 2014.

[5] A. Komianos, "The autonomous shipping era. operational, regulatory, and quality challenges," *TransNav: International Journal on Marine Navigation and Safety of Sea Transportation*, vol. 12, no. 2, 2018.

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer, 2016, pp. 21–37.

[8] RangiLyu, “NanoDet-Plus: Super fast and high accuracy lightweight anchor-free object detection model.” https://github.com/RangiLyu/nanodet, 2021.

[9] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama *et al.*, “Speed/accuracy trade-offs for modern convolutional object detectors,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7310–7311.

[10] Ø. J. Rødseth and H.-C. Burmeister, “Risk assessment for an unmanned merchant ship,” *TransNav: International Journal on Marine Navigation and Safety of Sea Transportation*, vol. 9, no. 3, pp. 357–364, 2015.

[11] M. Ahmed, A. B. Bakht, T. Hassan, W. Akram, A. Humais, L. Seneviratne, S. He, D. Lin, and I. Hussain, “Vision-based autonomous navigation for unmanned surface vessel in extreme marine conditions,” in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 7097–7103.

[12] S. Dodge and L. Karam, “Understanding how image quality affects deep neural networks,” in *2016 8th International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, 2016, pp. 1–6.

[13] R. Geirhos, C. Temme, J. Rauber, H. Schütt, M. Bethge, and F. Wichmann, “Generalisation in humans and deep neural networks,” in *32nd Conference on Neural Information Processing Systems (NeurIPS 2018)*. Curran Associates, Inc., 2019, pp. 7538–7550.

[14] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, “Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness,” in *International Conference on Learning Representations*, 2018.

[15] D. Hendrycks and T. Dietterich, “Benchmarking neural network robustness to common corruptions and perturbations,” in *International Conference on Learning Representations*, 2019.

[16] C. Liu, Y. Dong, W. Xiang, X. Yang, H. Su, J. Zhu, Y. Chen, Y. He, H. Xue, and S. Zheng, “A comprehensive study on robustness of image classification models: Benchmarking and rethinking,” *arXiv preprint arXiv:2302.14301*, 2023.

[17] N. Mu and J. Gilmer, “MNIST-C: A robustness benchmark for computer vision,” *arXiv preprint arXiv:1906.02337*, 2019.

[18] A. Barbu, D. Mayo, J. Alverio, W. Luo, C. Wang, D. Gutfreund, J. Tenenbaum, and B. Katz, “ObjectNet: A large-scale bias-controlled dataset for pushing the limits of object recognition models,” in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d. Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019.

[19] D. Hendrycks, S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. Zhu, S. Parajuli, M. Guo *et al.*, “The many faces of robustness: A critical analysis of out-of-distribution generalization,” in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE Computer Society, 2021, pp. 8320–8329.

[20] C. Michaelis, B. Mitzkus, R. Geirhos, E. Rusak, O. Bringmann, A. S. Ecker, M. Bethge, and W. Brendel, “Benchmarking robustness in object detection: Autonomous driving when winter is coming,” *arXiv preprint arXiv:1907.07484*, 2019.

[21] X. Nie, M. Yang, and R. W. Liu, “Deep neural network-based robust ship detection under different weather conditions,” in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 47–52.

[22] J. Guo, H. Feng, H. Xu, W. Yu, and S. shuzhi Ge, “D3-Net: integrated multi-task convolutional neural network for water surface deblurring, dehazing and object detection,” *Engineering Applications of Artificial Intelligence*, vol. 117, p. 105558, 2023.

[23] Y. Li, J. Guo, X. Guo, K. Liu, W. Zhao, Y. Luo, and Z. Wang, “A novel target detection method of the unmanned surface vehicle under all-weather conditions with an improved YOLOV3,” *Sensors*, vol. 20, no. 17, p. 4885, 2020.

[24] G. Jocher, A. Chaurasia, and J. Qiu, “YOLO by Ultralytics,” 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[25] C. Li, “High quality, fast, modular reference implementation of SSD in PyTorch,” https://github.com/lufficc/SSD, 2018.

[26] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[27] E. Arani, S. Gowda, R. Mukherjee, O. Magdy, S. Kathiresan, and B. Zonooz, “A comprehensive study of real-time object detection networks across multiple domains: A survey,” *arXiv preprint arXiv:2208.10895*, 2022.

[28] Z. Shao, W. Wu, Z. Wang, W. Du, and C. Li, “Seaships: A large-scale precisely annotated dataset for ship detection,” *IEEE Transactions on Multimedia*, vol. 20, no. 10, pp. 2593–2604, 2018.

[29] D. K. Prasad, D. Rajan, L. Rachmawati, E. Rajabally, and C. Quek, “Video processing from electro-optical sensors for object detection and tracking in a maritime environment: A survey,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 8, pp. 1993–2016, 2017.

[30] R. Lahouli, G. D. Cubber, B. Pairet, C. Hamesse, T. Fréville, and R. Haelterman, “Deep learning based object detection and tracking for maritime situational awareness,” in *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2022, pp. 643–650.

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

[32] J.-H. Kim, N. Kim, Y. W. Park, and C. S. Won, “Object detection and classification based on YOLO-V5 with improved maritime dataset,” *Journal of Marine Science and Engineering*, vol. 10, no. 3, p. 377, 2022.