Benchmark: Object Detection for Maritime Search and Rescue

Dmitrii Kirov*, Simone Fulvio Rollini, Rohit Chandrahas, Shashidhar Reddy Chandupatla, and Rajdeep Sawant

Collins Aerospace

Abstract. We propose an object detection system for maritime search and rescue as a benchmark problem for verification of neural networks (VNN). The model to be verified is a YOLO (You Only Look Once) deep neural network for object detection and classification and has a very high number of learnable parameters (millions). We describe the workflow for defining and generating robustness properties in the regions of interest of the images, i.e., in the neighborhood of the objects to be detected by the neural network. This benchmark can be used to assess the applicability and the scalability of existing VNN tools for perception systems based on deep learning.

URL. Benchmark materials, such as trained models (.onnx), examples of properties (.vnnlib), test images, and property generation procedures, are available at `https://github.com/loonwerks/vnncomp2023`.

1 Introduction

Generally, maritime surveillance has been conducted using means such as satellites and manned aircraft. These have been limited in their ability to provide high-resolution, real-time video processing for various reasons. For example, satellite bandwidth is not ideal to handle large amounts of real time data, while rotorcrafts and fixed wing platforms can be very expensive. Current developments with quadcopters and other small drones have enabled additional options that are much cheaper than larger aircraft. Such unmanned vehicles can be equipped with various sensors and signal processing algorithms to enable automated identification of regions where search and rescue efforts may be focused, thereby reducing the time to rescue that can be very limited [1].

One enabling technology for this application is computer vision that is powered by state-of-the-art Machine Learning (ML) algorithms and provides reliable object detection and classification functions. Trustworthiness of these ML-based functions is of paramount importance, because their degraded performances and failures may significantly reduce the chances of people in distress to be rescued timely, thus impacting the safety. Therefore, *learning assurance* process and methods [4] need to be applied to guarantee the expected performance and the robustness of the ML-based search and rescue system. In particular, *formal*

* Corresponding author. Email: dmitrii.kirov@collins.com

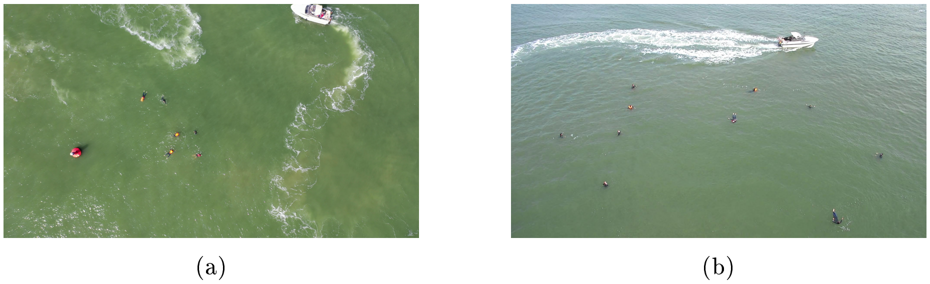
methods can be effective means of assessing the robustness of perception-based models, for example, to noise and other adverse inputs [3].

This benchmark problem intends to challenge the VNN community in the direction of verifying properties of large-scale neural networks for computer vision, such as object detection networks. It has also been submitted to the 2023 VNNComp event[1], along with few other benchmarks on YOLO neural networks. To the best of our knowledge, this is the first time when such benchmarks have been proposed for VNN tools.

2 Model Description

The object detection model chosen as a basis of the benchmark is a YOLOv5-nano neural network (NN). The objective is to assess current capabilities of VNN tools for verifying properties of deep neural networks of high complexity, such as YOLOs. The "nano" version of a YOLO has been selected for the sake of an incremental approach to complexity. It contains much fewer learnable parameters compared to its original, full-scale YOLO version. Furthermore, to make the model supported by VNN tools, SiLU activation functions have been replaced with (piecewise linear) Leaky ReLU activations.

The model has been trained on the SeaDronesSee dataset[2] [5] [2] consisting of maritime search and rescue scenarios captured using a drone, generated by the University of Tuebingen. Examples of images used in the benchmark are shown on Figure 1. The dataset includes of 8930 training images and 1547 validation images, all of which are labelled with bounding boxes and corresponding classes. There are six classes in the dataset, such as "boat", "person", "jetski". The intended function of the YOLO model is to detect objects of these classes on the water surface, draw bounding boxes around them, and classify them. Model outputs can be communicated to the operator at the ground station who can use this information to dispatch rescue missions and vehicles.


(a)

(b)

Fig. 1: Examples of images used to formalize benchmark properties.

[1] https://sites.google.com/view/vnn2023

[2] SeaDronesSee dataset is used on a license.

The YOLOv5-nano model has 157 layers and around 1.8×10^6 learnable parameters. The training has been done with an input image size of 640×640. The model has 3 output layers. The total input and output size of the model are, respectively, 1.2×10^6 and 277×10^3 (note that the numbers have been rounded).

3 Property Description

This benchmark focuses on robustness properties that are formalized using L_∞ norms, same as in many other existing benchmarks and applications. Such evaluation is an important starting point in assessing the applicability of VNN tools to object detection models, such as YOLO.

3.1 Robustness properties overview

Robustness properties are formulated by applying perturbations to selected inputs and requiring that the predicted class remains unchanged, while allowing for a bounded decrease of the confidence of object existence. The motivation is to keep the object detector robust, for example, to different lighting conditions and, possibly, to adversarial attacks. Robustness is particularly relevant to the detection of swimmers on the water surface, because misclassifications and false negatives can have a significant safety impact: for example, a person could not get noticed and, as a consequence, not rescued or rescued too late. The benchmark represents local robustness by applying L_∞ perturbations to the neighborhood of objects (e.g., persons, boats) in the image[3]. The neighborhood is determined from the bounding box, which corresponds to the model detection of the object on the original unperturbed image.

Mathematically, local robustness properties are defined in the “delta-epsilon” form, which makes use of the infinity norm:

$$||x' - x||_\infty < \delta \implies ||f(x') - f(x)||_\infty < \epsilon. \tag{1}$$

where $x \in X$ is the original input (image) belonging to the input space X of the ML model, $x' \in X$ is the perturbed input, $f(x)$ and $f(x')$ are ML model outputs for, respectively, x and x', δ and ϵ are as discussed above ($\delta, \epsilon \in \mathbb{R}_{>0}$), and $|| \cdot ||$ is a norm that measures the distance between original and perturbed inputs and outputs. Local robustness requires that for an input perturbation bounded by δ (precondition $||x' - x||_\infty < \delta$) the output must not deviate by more than ϵ (postcondition $||f(x') - f(x)||_\infty < \epsilon$).

[3] We note that in future work it may be possible to also identify more meaningful perturbations, such as changing the colors of certain objects in the image (e.g., life jackets from red to blue). Such modifications may require additional image processing to precisely identify the pixels to apply perturbation to, which brings additional challenges to be solved.

3.2 Robustness properties for the YOLO model

Current benchmark includes robustness properties for different pixel perturbation magnitudes δ, ranging from 0.1% to 10%. Pixel perturbations are applied in the neighborhood of objects on the water surface (e.g., swimmers, boats) in order to see whether their detection changes due to modification in (or near) the pixels belonging to the object. The following steps are executed to formalize a robustness property, for a given δ:

1. Randomly pick a dataset image x;
2. Downscale[4] the image (including padding, if necessary) to match the NN input size (640×640), obtaining image x_{in};
3. Perform NN inference on the downsized image to compute bounding boxes $b_i^{out} \in B^{out}$ and respective classes ($1 \leq i \leq |B^{out}|$, where B^{out} is the set of bounding boxes predicted for the image obtained after post-processing of the NN output);
4. Upscale bounding boxes to the original image size, getting the boxes $b_i \in B$ ($1 \leq i \leq |B|$, considering that $|B| = |B^{out}|$);
5. Randomly pick one bounding box b from B, corresponding to one of the objects detected on the image;
6. Define a space of possible perturbations on the original-size image x, by applying L_∞ to pixels inside the bounding box b, and downscale it to the NN input size (see next section for details). Impose input constraints that the input perturbation is within δ, i.e., pixel perturbations are bounded by $x - \delta$ and $x + \delta$ for the original size image and by $(x - \delta)_{in}$ and $(x + \delta)_{in}$ for the downscaled image;
7. Impose output constraints that (1) the bounding box class confidence on the perturbed image is still the highest among all classes and that (2) the object existence confidence does not deviate by more than $\epsilon = 10\%$.

In this procedure, input constraints correspond to the precondition of the local robustness property, while output constraints correspond to its postcondition.

Key challenge. The main challenging aspect of the benchmark is the large number of inputs and outputs (on the order of, respectively, 10^6 and 10^5). The former is due to the need of using a high-resolution image in the search and rescue application, because some objects, such as swimmers, are often very small (e.g., due to high flight altitudes of the drone). Therefore, further decrease of the input size of the YOLO model significantly hampers its prediction performance.

4 Property Generation

The formulation of the properties required to address two technical issues. The first one is the mismatch between the size of original dataset images and the NN input size. To encode the input constraints, perturbation range for the pixels

[4] Resizing is performed via OpenCV `resize()` command (bilinear interpolation).

belonging to the selected area/bounding box is to be set to the range between $x-\delta$ and $x+\delta$. Since the selected bounding box b^{out} computed by the NN refers to the downscaled image x_{in}, b^{out} is first upsized to b on the original image x; then the perturbation is applied within b, leading to pixel perturbation range between $x-\delta$ and $x+\delta$, leaving the area of x outside of b unmodified. Finally, the images containing the two perturbed bounding boxes are downscaled to the NN input size as $(x-\delta)_{in}$ and $(x+\delta)_{in}$.

The second issue depends on the postprocessing applied to the NN output. The output bounding boxes and their respective classes are derived from the NN "raw" output of size $\sim 25200 \times 11$, consisting of candidate boxes and probability estimates, by means of a non-trivial sequence of operations, including multiple phases of threshold-based filtering and the application of a Non-Maximum Suppression algorithm to resolve overlaps. To encode the output constraints, which are expressed in terms of the NN raw output, an algorithm has been implemented to trace the selected bounding box back to the corresponding raw data.

5 Concluding Remarks

Robustness assessment of vision-based systems, such as object detection, is one of their key certification objectives. European Union Aviation Safety Agency (EASA) in their Concept Paper for Level 1&2 Machine Learning Applications [4] emphasizes Formal Methods (FM) as anticipated means of compliance for the verification of robustness of ML models. FM tools can become a critical enabler of AI/ML trustworthiness [3], therefore, aviation industry is looking forward to maturation and improvement of relevant technologies. The proposed benchmark has the intent of being a motivating example of a realistic application in the aerospace domain.

Acknowledgements. The authors wish to thank Benjamin Kiefer et al. from the University of Tuebingen for the publication of the *SeaDronesSee* dataset, which has been used as a baseline in this research work.

References

1. Avalon - Aerial and vision-based assistance system for real time object detection in search and rescue missions [Online]. https://uni-tuebingen.de/fakultaeten/mathematisch-naturwissenschaftliche-fakultaet/fachbereiche/informatik/lehrstuehle/kognitive-systeme/projects/avalon/
2. SeaDronesSee dataset [Online]. https://seadronessee.cs.uni-tuebingen.de/dataset
3. EASA and Collins Aerospace: Formal Methods use for Learning Assurance (ForMuLA). Tech. rep. (April 2023)
4. European Union Aviation Safety Agency (EASA): Concept Paper: Guidance for Level 1&2 Machine Learning Applications. Concept paper for consultation. (February 2023)
5. Varga, L.A., Kiefer, B., Messmer, M., Zell, A.: Seadronessee: A maritime benchmark for detecting humans in open water. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. pp. 2260–2270 (2022)