Volume 44 http://acousticalsociety.org/

6th Underwater Acoustics Conference & Exhibition

20-25 June 2021

Underwater Acoustics: Towards Automatic Target Recognition. Detection, Classification and Modelling

A study on modern deep learning detection algorithms for automatic target recognition in sidescan sonar images

Yannik Steiniger
Department of Situational Awareness, German Aerospace Center DLR Institute for the Protection of Maritime Infrastructures, Bremerhaven, Bremen, 27572, GERMANY; yannik.steiniger@dlr.de

Johannes Groen
Atlas Elektronik GmbH, Bremen, 28309, GERMANY; Johannes.Groen@atlas-elektronik.com

Jannis Stoppe
German Aerospace Center DLR Institute for the Protection of Maritime Infrastuctures, Bremerhaven, 27562, GERMANY; Jannis.Stoppe@dlr.de

Dieter Kraus
City University of Applied Sciences: Hochschule Bremen, Bremen, 28199, GERMANY; dieter.kraus@hs-bremen.de

Tobias Meisen
University of Wuppertal: Bergische Universitat Wuppertal, Wuppertal, North Rhine-Westphalia, 42119 GERMANY; meisen@uni-wuppertal.de

State-of-the art deep learning models have shown remarkable performance on computer vision tasks like object classification or detection. These networks are typically trained on large-scale datasets of natural RGB images. However, sidescan sonar images are gray-scaled images representing acoustic intensities. The fundamental differences between camera and sonar as well as the images itself makes it necessary to investigate the transfer of results achieved on RGB images to the sonar imagery domain. Therefore, we compare the deep learning detection algorithm YOLOv2 with its updated version YOLOv3, both adopted for object detection in sidescan sonar images. In addition to this, a small convolutional neural network (CNN) is trained from scratch and used for detection. The experiments answer two questions: First, whether, as for general computer vision problems, transfer learning of large deep learning models is preferable over training of custom networks when dealing with limited sonar data. Secondly, whether improvements in the YOLO architecture, developed based on RGB images, lead to significant improvements on sonar data as well. Our results show that YOLOv3 indeed performs better than YOLOv2. Furthermore, YOLOv3 achieves a true positive rate of up to 98.2% and outperforms the small CNN.

Published by the Acoustical Society of America

 https://doi.org/10.1121/2.0001470

1. INTRODUCTION

Inspection of the seafloor is a non-trivial task that is usually performed, from a hardware perspective, using imaging sonar systems such as sidescan or synthetic aperture sonars. These sonars are usually mounted on an autonomous underwater vehicle (AUV) in order to scan a large area autonomously. Possible applications are for example pipeline tracking or the detection of unexploded ordnance. In order to process the large amount of data captured during a mission, so-called automatic target recognition (ATR) algorithms have been developed. These algorithms are typically based on a template matcher for a first detection and feature extractor and classifier for further differentiation of the detections [4].

In the areas of detection and classification, deep learning algorithms have shown high potential compared to traditional approaches [8, 12, 13], leading us to the conclusion that it is reasonable to investigate the potential of these methods within ATR as well. Those networks are typically trained on large-scale datasets of natural RGB images like ImageNet [14] or MS COCO [9]. However, sidescan sonar images are gray-scaled images representing acoustic intensities. Due to the fundamental differences between camera and sonar as well as RGB and sonar image, it is not straight forward that recent achievements in computer vision (based on RGB images) also apply to sonar data.

While most research in the field of deep learning for sonar imagery focuses on the classification task [2, 16, 17], there are only few other works tackling the detection problem. Einsidler et al. were the first applying YOLOv2 [11] to sidescan sonar images in order to detect rocks and anomalies [3]. Other works use a RetinaNet [10] to detect rocks [5,6] in sidescan mosaic images. An architecture called MiNet which is similar to YOLO was developed in [15] for the detection of mine like objects. Berthomier et al. transformed a convolutional neural network (CNN) trained for classification of target and clutter snippets in order to apply it as a detector [1].

In this paper, we address two essential questions, not answered in the previously mentioned publications. First, whether transfer learning of large deep learning detector is preferable over training of a smaller custom network especially if only a small dataset of sonar images is available. Secondly, whether improvements in the deep learning architectures, which were developed based on RGB images, lead to significant improvements on sonar data as well. To answer these questions, we compare the deep learning detection algorithm YOLOv2 [11] with its updated version YOLOv3 [12], both adopted for object detection in sidescan sonar images. In addition to this, a small CNN is trained from scratch and used for detection. We compare the detection performance based on the receiver operating characteristic (ROC) curves and run additional experiments to further investigate the behavior of the three approaches.

The remaining of the paper is organized as follows. In Section 2 the data as well as the considered algorithms and training routine are presented. Following this, Section 3 answers the two aforementioned research questions by carrying out several experiments. Finally, a conclusion and outlook is given in Section 4.

2. MATERIALS AND METHODS

A. DATASET

Over the course of several sea and harbor expeditions, sidescan sonar data were collected with an Edgetech 2205 sidescan sonar mounted on a SeaCat AUV (ATLAS ELEKTRONIK GmbH, Bremen, Germany). The sidescan sonar operates at a center frequency of 850 kHz, with a bandwidth of 45 kHz. An experimental signal processing chain produces gray-scaled images in a waterfall manner, i.e., the individual signals of each ping period are consecutively stacked on top of each other. These waterfall images have a pixel resolution of 10 cm in along and across the track direction. Figure 1 shows an example of a processed waterfall image. For the later processing by the deep learning architectures the waterfall images are split into slightly overlapping squared images.

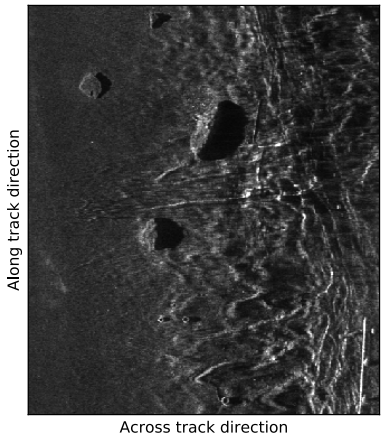


Figure 1: Example of a waterfall sidescan sonar image.

Table 1: Overview about the datasets.

Mission	Number of images	Number of targets	Dataset
Harbour 1	70	95	training
Sea 1	182	575	training
Sea 2	517	1661	training
Harbour 2	5	5	training
Lake 1	422	55	testing

In order to generate a training and test dataset, objects present in the images are labeled with a bounding box, which encloses the object's highlight and shadow. Objects labeled in this way include rocks, cylinders, tires and wrecks. All objects are combined in one class *target*. Table 1 gives an overview about the dataset. Images from the missions Harbour 1&2 and Sea 1&2 are used for training while data from Lake 1 is used for testing. As can be seen from the number of images and targets in the test dataset, several images contain no object in order to be able to investigate the false alarm rate for those cases.

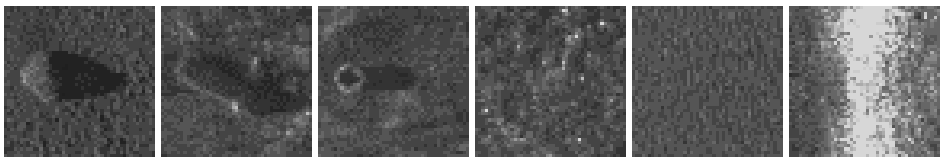
For training of the small CNN, as described in the next section, the snippets showing the objects are extracted from the waterfall images. If $(x_{1,b}, y_{1,b})$ and $(x_{2,b}, y_{2,b})$ are the corner points of a ground truth bounding box, then the corner points

$$(x_{1,s}, y_{1,s}) = \left(x_{1,b}, y_{1,b} + \left\lfloor \frac{y_{2,b} - y_{1,b}}{2} \right\rfloor - 23 \right) \tag{1}$$

and

$$(x_{2,s}, y_{2,s}) = \left(x_{1,b} + 46, y_{1,b} + \left\lfloor \frac{y_{2,b} - y_{1,b}}{2} \right\rfloor + 23 \right) \tag{2}$$

are used to extract snippets of size 47×47. Additionally, 2336 snippets at random locations are extracted and labeled as a class *clutter*. By masking the locations of target snippets, we ensure that the clutter snippets do not contain parts of the objects. Some examples of snippets from the class *target* and *clutter* are shown in Fig. 2. All pixels in the waterfall images are clipped to the interval $[0.1, 5]$, in order to increase the contrast in the image. Prior to extracting the snippets, the pixel values are linearly scaled to the interval $[-1, 1]$. Images from the port-side sonar are flipped such that the shadow of an object always lies on the right side of the object. During training, the dataset is augmented using horizontal flipping, since vertical flipping would cause the shadow to lie on the left side of the object again.



(a) *(b)* *(c)* *(d)* *(e)* *(f)*

Figure 2: Example of extracted target (a)-(c) and clutter (d)-(f) snippets for pre-training of the small CNN.

B. DEEP LEARNING OBJECT DETECTION

We compare three deep learning architectures for purpose of detecting the labeled targets in the sonar images. The first two are the well-known YOLOv2 and YOLOv3, which were pre-trained on the MS COCO [9] dataset. Both networks are used with an input size of 608×608. For training the waterfall images are resized to this required size using bilinear interpolation. The transfer learning is carried out for 100 epochs with the Adam optimizer [7] and a batch size of 64 for YOLOv2 and 32 for YOLOv3. The learning rate is set to 0.001 for the first 50 epochs and then reduced by a factor 10.

The third architecture is a small CNN adapted from the work of Berthomier et al. [1]. Here a CNN, whose structure is described in Table 2, is first trained on a classification task. The CNN should learn to distinguish the *target* from the *clutter* snippets. In this pre-training the input size of the CNN is 47×47. In order to use this CNN for detection, the input and output layers need to be adapted. The input size is expanded to 600×600, which results in an output feature map of shape $35 \times 35 \times 4$ after the last max-pooling layer. The dense layer is replaced by a 1×1 convolution, combining the four channels of the output feature map into one localization map. The weights in this 1×1 convolution layer are used from the pre-trained dense layer. With this modification the CNN is now a fully convolutional network (FCN) with a receptive field of 47×47 and can be seen as the small CNN for classification applied to the waterfall image in a convolutional manner. The pixel value in the localization map can be seen as a confidence score for the detection. Pixel with a high value correspond to positions in the input image where the snippet represented by the current receptive field would produce a high probability of containing a target. Since the overall stride of the small CNN for detection is smaller than its receptive field, the presence of an object in the input image would lead to multiple pixels in the localization map having a high value. Thus, we use the suppression method described in [1], which sorts the detections based on their confidence score, masks a region around a detection in the input image and ignores detections of lower confidence in those regions.

The small CNN is trained on the classification task for 20 epochs. We choose the Adam optimizer with a learn rate of 0.001 and a batch size of 32. After training and the discussed modification for detection the small CNN outputs bounding boxes of size 47×47. This is different from other deep learning based detectors whose predicted bounding boxes depend on the size of the object (and shadow). The effect of this will be analyzed in the experiments in Section 3. Table 3 summarizes the comparison of the three architectures considered in this work.

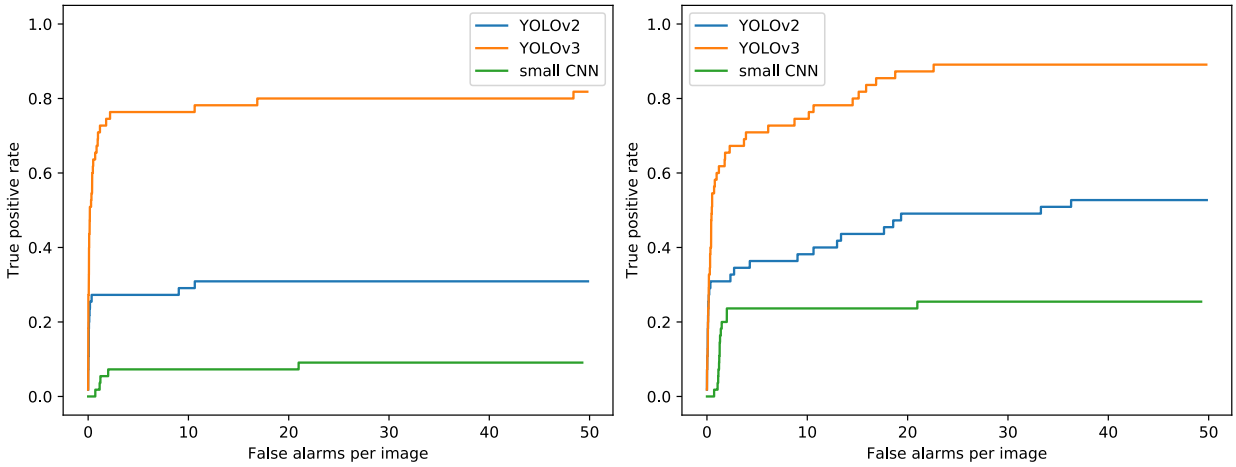
After training the three deep learning detectors their performance on the held-out test dataset Lake 1 is compared by means of ROC-like curves. Note that for detection the false positive rate can not be calculated because there are no true negative cases. Thus, we display the average number of false alarms per image on the abscissa and refer to these curves as ROC-like. The curves are generated by varying the detection threshold for the confidence score of the detectors and counting the number of true positive detections and false alarms in all test images. Since the small CNN only predicts bounding boxes of the same size, not only the intersection over union (IoU) is considered to determine true positive detections but also the pixel-wise Euclidean distance between the center pixel of the true and predicted bounding box.

Table 2: Architecture of the small CNN used for classification.

Layer	**Neurons**	**Kernel Size**	**Stride**	**Padding**	**Activation Function**	**Feature Map Dimension**
Input	-	-	-	-	-	$47 \times 47 \times 1$
Conv2D	4	4×4	$(1, 1)$	$(1, 1)$	relu	$44 \times 44 \times 4$
MaxPool2D	-	2×2	$(2, 2)$	-	-	$22 \times 22 \times 4$
Conv2D	4	3×3	$(1, 1)$	$(1, 1)$	relu	$20 \times 22 \times 4$
MaxPool2D	-	2×2	$(2, 2)$	-	-	$10 \times 10 \times 4$
Conv2D	4	3×3	$(1, 1)$	$(1, 1)$	relu	$8 \times 8 \times 4$
MaxPool2D	-	2×2	$(2, 2)$	-	-	$4 \times 4 \times 4$
Conv2D	4	3×3	$(1, 1)$	$(1, 1)$	relu	$2 \times 2 \times 4$
MaxPool2D	-	2×2	$(2, 2)$	-	-	$1 \times 1 \times 4$
Dense	1	-	-	-	sigmoid	1

Table 3: Comparison of the considered deep learning detectors.

Detector	**YOLOv2**	**YOLOv3**		**small CNN**
			classification	detection
Backbone	Darknet-19	Darknet-53	CNN-4	CNN-4
Parameter	50,547,678	61,523,724	517	517
Input	608×608	608×608	47×47	600×600
Output	$19 \times 19 \times 5 \cdot 7$	$\{19 \times 19 \times 3 \cdot 7,$ $38 \times 38 \times 3 \cdot 7,$ $76 \times 76 \times 3 \cdot 7\}$	1	35×35
Pre-training	MS COCO	MS COCO	-	classification
Bounding boxes	variable	variable	-	47×47



(a) $d_{IoU} = 0.5$ ***(b)*** $d_{dist} = 10\ px$

Figure 3: Comparison of the ROC-like curves of the three detectors.

3. RESULTS

In order to answer the question whether transfer learning of large state-of-the-art deep learning detectors on sidescan sonar images is preferable over training a small network for this task, Fig. 3 compares the ROC-like curves from YOLOv2, YOLOv3, and the small CNN on the test dataset. Note that we have repeated this and the following experiments three times to account for the randomness when initializing the small CNN or the new last layer of the YOLO detectors. In all runs we observed similar results. A minimum IoU of $d_{IoU} = 0.5$ and a maximum distance of the bounding box center points of $d_{dist} = 10$ px is selected to determine the true positives. In both cases YOLOv2 and YOLOv3 perform better than the small CNN. However, only YOLOv3 is able to detect at least 80% of the targets. When d_{dist} is considered for determining the true positives, the three detectors show a better performance compared to the ROC-like curves for d_{IoU}. Because the IoU takes also the shape of the bounding boxes into account, this is a more restrictive measure. The better performance for d_{dist} shows that many detections especially from YOLOv2 and the small CNN are placed correctly but the shape of the bounding box does not match the labeled one. With $d_{dist} = 10$ px, which correspond to a distance of 1 m, YOLOv3 detects 89.0% of the targets, YOLOv2 52.7% and the small CNN only 25.5%. These results show that using state-of-the-art detectors and transfer them on sonar imagery is preferable over training of a small CNN from scratch. Nevertheless, if for an application a higher true positive rate is required, e.g. >90% for detecting unexploded ordnance, more training data is required to improve the performance.

If the maximum distance between the centers of the ground truth and predicted bounding box is set to $d_{dist} = 20$ px and thus not as restrictive as before, the performance of the small CNN improves as shown in Fig. 4. As expected, all three detectors show better ROC-like curves. The largest improvement can be observed for the small CNN. Here a true positive rate of 0.836 can be reached and in the area of few false alarms per image the performance is comparable to the one of YOLOv3. This shows that the small CNN is able to detect many targets but the localization is not as accurate as with YOLOv3. If more uncertainty in the localization is allowed by using $d_{dist} = 20$ px, YOLOv3 is able to detect 98.2% of the targets.

To give a further insight on the detection behavior of the three methods, Fig. 5 and 6 show two example images from the test dataset with the corresponding ground truth and detections. The detection threshold is set to 0.1 in all cases. The first example shows two targets which lie very closely together. YOLOv2 is only able to detect one of them. One false alarm is generated due to a second bounding box originating from the

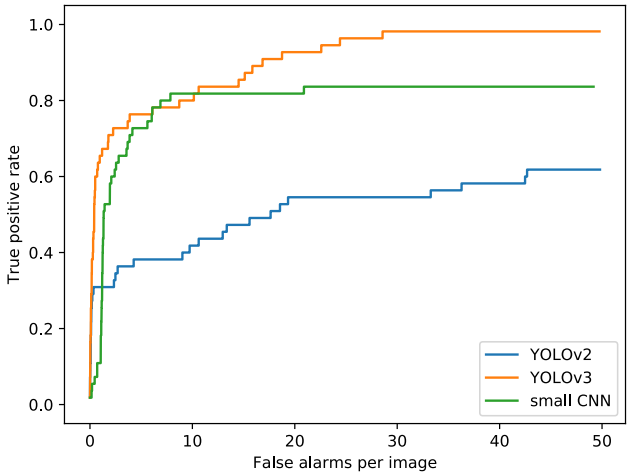
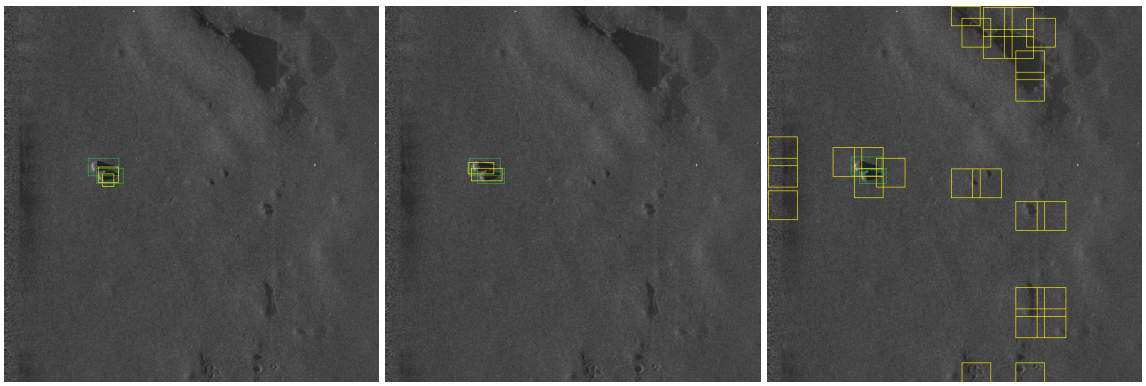


Figure 4: Comparison of the ROC-like curves of the three detectors for $d_{dist} = 20\ px$***.***



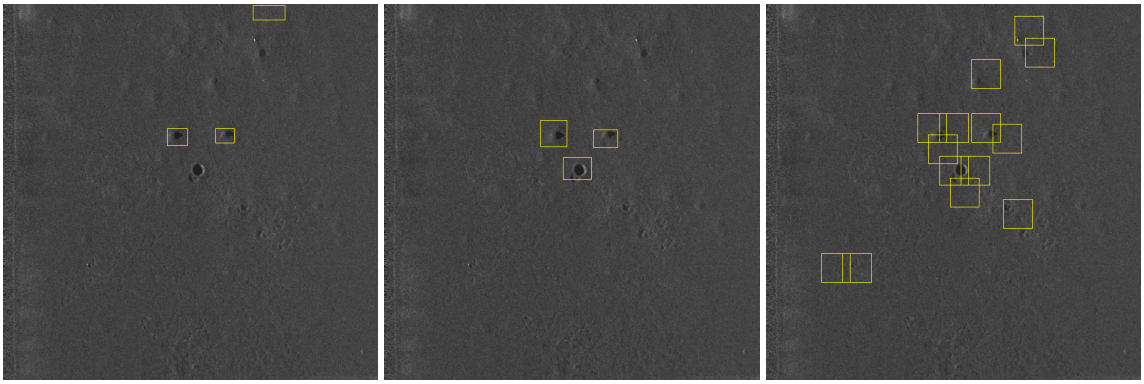
(a) YOLOv2 detections ***(b) YOLOv3 detections*** ***(c) Small CNN detections***

Figure 5: Detections results for an example image with two targets present and a detection threshold of 0.1. Ground truth in green and detection in yellow.

lower target, which encloses only a small portion of the object. With YOLOv3 both targets are detected and no false alarms are generated. The small CNN is able to detect both targets as well but generates a large number of false alarms. These false alarms are mainly located in regions of the image where a shadow is present, e.g. due to elevations or holes.

In the second example no target is present. However, all three detectors output detections. The two hillocks in the middle of the image show some highlight and shadow characteristic which looks similar to the targets in the training dataset. Due to this, all three algorithms are falsely detecting them as a target. In addition, YOLOv3 generates a false alarm for the hole in the middle of the image. Looking at more test images, this exposes as a main source of false alarms for YOLOv3. Compared to the YOLO-based detectors, the small CNN is again producing more false alarms. Similar to the first example at many locations multiple false alarms are generated. Because the CNN is shifted over the image in a convolutional manner, the presence of a target or target-like structure results in multiple active positions in the localization map. An increase in the size of the mask in the suppression algorithm could reduce the number of false alarms but at a risk of not being able to detect two close targets as in Fig. 5.

In contrast to YOLOv2 and YOLOv3, the small CNN was not trained on the whole waterfall image but only on snippets which can be seen as a subset of the information available to the other two detectors. Thus,



(a) YOLOv2 detections *(b) YOLOv3 detections* *(c) Small CNN detections*

Figure 6: Detection results for an example with no targets present and a detection threshold of 0.1.

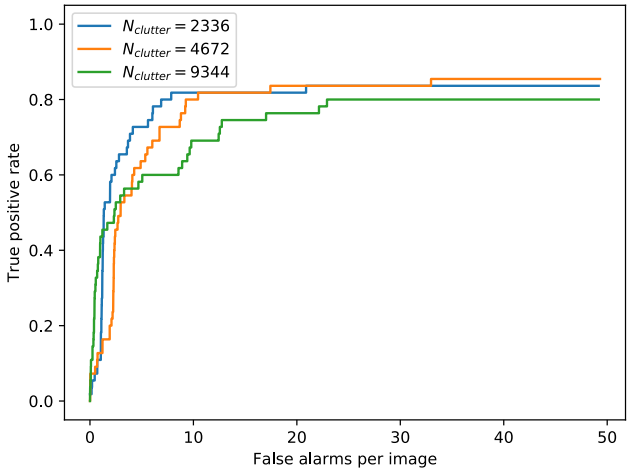


Figure 7: Comparison of the ROC-like curves of the small CNN for different number of clutter training snippets and $d_{dist} = 20\ px$***.***

Fig. 7 investigates the effect of considering more clutter samples during the training of the small CNN. Taking more samples from the class *clutter* into account gives the CNN more information especially about the background, which may reduce the number of false alarms. However, because the number of target snippets is limited this results in an unbalanced training set, which can harm the training of the CNN. In Fig. 7 it can be seen that adding more clutter samples to the training does not have a positive impact on the detection performance. Both CNNs trained with more clutter snippets either detect fewer targets or produce more false alarms. Further methods like specific class weights in the loss function or more augmentation of the target snippets may compensate the negative effect of the unbalanced dataset but were not considered in this work.

The difference in performance between YOLOv2 and YOLOv3 is certainly interesting to investigate. It is not to be expected in this form, as YOLOv3 performs better on benchmarks like MS COCO, but not necessarily to such an extent. One modification in YOLOv3 compared to YOLOv2 is the detection at multiple scales which should improve the performance for small objects [12]. In order to investigate if this also applies to sonar data, we split our test dataset in half, taking into account the size of the ground truth bounding boxes. There are 27 small objects with a bounding box area between 234 px^2 and 897 px^2 as well as 28 large objects with an area between 943 px^2 and 6588 px^2. On both subsets the performance of

Table 4: Comparison of the inference speed of the three detectors.

Detector	Number of parameter	Inference speed in FPS	
		CPU	GPU
YOLOv2	50,547,678	0.18	53
YOLOv3	61,523,734	0.21	31
small CNN	517	44	500

YOLOv3 is better than the one of YOLOv2. There is also no drop in performance for smaller objects in YOLOv2 observable. YOLOv2 detects nine small and eight large objects, while YOLOv3 detects 22 and 23 respectively. Thus, the improvements made in YOLOv3, like the detection at multiple scales or the deeper backbone, apply to both kinds of objects.

Since for some applications, e.g. on-board processing during a mission, the computation time becomes relevant, Table 4 lists the inference speed of the three detectors measured on a Intel Xeon W-2155 CPU and on a Nvidia Quadro P6000 GPU in frames per second (FPS). Due to the smaller model size the small CNN is faster than both YOLO-based detectors. Nevertheless, YOLOv2 and YOLOv3 run with more than 30 FPS which can be considered in real-time for many applications. In an on-board scenario where a ping rate of 10 Hz is considered, capturing 600 pings and processing them to form a waterfall image would take more than 60 s, which is less than the time it takes for YOLOv3 to process one image on the GPU. However, in this scenario the algorithm will probably not run on a high-end GPU but on some small device like a Jetson Nano on which the algorithm will run slower. The investigations made in this work show promising results but the actual application of the detectors to an on-board scenario still needs to be studied in more depth.

4. CONCLUSION

In this work YOLOv2, YOLOv3 and a detector based on a small CNN have been compared for the task of detecting targets in sidescan sonar images. The experiments have shown that improvements in the YOLO architecture from YOLOv2 to YOLOv3, which were developed based on RGB image data, also improve the performance on sonar data. In all experiments the newer YOLOv3 showed better ROC-like curves then YOLOv2. The comparison with the small CNN gives answers to the question whether transfer learning of large deep learning models for detection is preferable over training of a custom network from scratch when dealing with limited sonar image data. Only when being less strict about the distance error between prediction and ground truth, the small CNN is able to outperform YOLOv2, while YOLOv3 still performs the best with a true positive rate of 0.982. This leads to the answer that, as for general computer vision problems, transfer learning large models is preferable when applying deep learning to the task of ATR in sidescan sonar images.

An interesting finding is the observed plateau in the ROC-like curves that is quickly reached for all the methods. The plateau is sufficiently high for YOLOv3, but this is not the case for the other two methods. This is essential, since the typically required true positive rate is higher than 0.9. Whether this inability to detect certain objects can be explained or overcome by better training is part of future work. We also plan to investigate more deep learning based detectors for the task of detecting targets in sidescan sonar imagery and to compare them to traditional methods.

ACKNOWLEDGMENTS

The authors would like to thank the ATLAS ELEKTRONIK GmbH for their support of this work.

REFERENCES

[1] T. Berthomier; D. P. Williams; S. Dugelay, "Target Localization in Synthetic Aperture Sonar Imagery using Convolutional Neural Networks", In Proceedings of the OCEANS 2019 MTS/IEEE Seattle, Seattle, WA, USA, 27-31 October 2019, pp. 1-9.

[2] A. Bouzerdoum; P. B. Chapple; M. Dras; Y. Guo; L. Hamey; T. Hassanzadeh; H. T. Le; O. Nezami; M. Orgun; S. L. Phung; C. H, Ritz; M. Shahpasand, "Improved Deep-Learning-based Classification of Mine-like Contacts in Sonar Images from Autonomous Underwater Vehicle", In Proceedings of the 5th Underwater Acoustics Conference and Exhibition (UACE), Crete, Greece, June 30 - July 05 2019, pp. 179-186.

[3] D. Einsidler; M. Dhanak; P.-P. Beaujean, "A Deep Learning Approach to Target Recognition in Side-Scan Sonar Imagery", In Proceedings of the OCEANS 2018 MTS/IEEE Charleston, Charleston, SC, USA, 22-25 October 2018, pp. 1-4.

[4] T. Fei; D. Kraus; A. M. Zoubir, "Contributions to Automatic Target Recognition Systems for Underwater Mine Classification", *IEEE Transactions on Geoscience and Remote Sensing* **53** (1), 505-518 (2015).

[5] P. Feldens; A. Darr; A. Feldens; F. Tauber, "Detection of Boulders in Side Scan Sonar Mosaics by a Neural Network", *Geosciences* **9** (4), 159 (2019).

[6] P. Feldens, "Super Resolution by Deep Learning Improves Boulder Detection in Side Scan Sonar Backscatter Mosaics", *Remote Sensing* **12** (14), 2284 (2020).

[7] D. P. Kingma; J. Ba, "Adam: A Method for Stochastic Optimization", In Proceedings of the 3rd International Conference on Learning Representations, San Diego, CA, USA, 7-9 May 2015; Y. Bengio, Y. LeCun, Eds.; 2015.

[8] A. Krizhevsky; I. Sutskever; G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", In *Advances in Neural Information Processing Systems 25*; R. Pereira, C. J. C. Burges, L. Bottou, K. Q. Weinberger, Eds.; MIT Press; **1**, 1097-1105 (2012).

[9] T.-Y. Lin; M. Maire; S. Belongie; J. Hays; P. Perona; D. Ramanan; P. Dollár; C. L. Zitnick, "Microsoft COCO: Common Objects in Context", *Computer vision - ECCV 2014*; Springer **8693**, 740-755 (2014).

[10] T.-Y. Lin; P. Goyal; R. Grishick; K. He; P. Dollár, "Focal Loss for Dense Object Detection", *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, 20-29 October 2017, pp. 2999-3007.

[11] J. Redmon; A. Farhadi, "YOLO9000: Better, Faster, Stronger", *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 21-26 June 2017, pp. 6517-6525.

[12] J. Redmon; A. Farhadi, "YOLOv3: An Incremental Improvement", *arXiv:1804.02767*, 2018, [online] Available: https://arxiv.org/abs/1804.02767v1.

[13] S. Ren; K. He; R. Girshick; J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", In *Advances in Neural Information Processing Systems 28*; C. Cortes, N. D. Lawrence, D. Lee, M. Sugiyama, R. Garnett, Eds.; MIT Press; **1**, 91-99 (2015).

[14] O. Russakovsky; J. Deng; H. Su; J. Krause; S. Satheesh; S. Ma; Z. Huang; A. Karpathy; A. Khosla; M. Bernstein; A. C. Berg; L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge", *International Journal of Computer Vision* **115**, 211-252 (2015).

[15] J. M. Topple; J. A. Fawcett, "MiNet: Efficient Deep Learning Automatic Target Recognition for Small Autonomous Vehicles", *IEEE Geoscience and Remote Sensing Letters* **18** (6), 1014-1018 (2021).

[16] D. P. Williams, "Underwater Target Classification in Synthetic Aperture Sonar Imagery Using Deep Convolutional Neural Networks", In Proceedings of the 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 4-8 December 2016, pp. 2497-2502.

[17] D. P. Williams, "On the Use of Tiny Convolutional Neural Networks for Human-Expert-Level Classification Performance in Sonar Imagery", *IEEE Journal of Oceanic Engineering* **46** (1), 236-260 (2021).