Convolutional Neural Network Architectures for Sonar-Based Diver Detection and Tracking

Igor Kvasić, Nikola Mišković, Zoran Vukić
Laboratory for Underwater Systems and Technologies
Faculty of Electrical Engineering and Computing
University of Zagreb
Zagreb, Croatia
e-mail: igor.kvasic@fer.hr, nikola.miskovic@fer.hr, zoran.vukic@fer.hr

***Abstract*—Autonomous underwater navigation presents a whole set of challenges to be resolved in order to become adequately accurate and reliable. That is particularly critical when human divers work in close collaboration with autonomous underwater vehicles (AUVs). In absence of global positioning signals underwater, acoustic based sensors such as LBL (long-baseline), SBL (short-baseline) and USBL (ultrashort-baseline) are commonly used for navigation and localization. In addition to these low-bandwidth and high latency technologies, cameras and sonars can provide position measurements relative to the vehicle which can be used as an aid for navigation as well as for keeping a safe working distance between the diver and the AUV. While optical cameras are highly affected by water turbidity and lighting conditions, sonar images often become hard to interpret using conventional image processing methods due to image granulation and high noise levels.**

This paper focuses on finding a robust and reliable sonar image processing method for detection and tracking of human divers using convolutional neural networks. Machine learning algorithms are making a huge impact in computer vision applications but are not always considered when it comes to sonar image processing. After presenting commonly used image processing techniques the paper will focus on giving an overview of state-of-the-art machine learning algorithms and explore their performance in custom sonar image dataset processing. Finally, the performance of these algorithms will be compared on a set of sonar recordings to determine their reliability and applicability in a real-time operation.

Keywords— Convolutional neural networks, object detection, diver detection, sonar image processing

I. Introduction

The use of underwater robots can greatly facilitate human tasks and improve safety conditions in harsh and unstructured environments where the slightest unexpected malfunction or disturbance can lead to catastrophic events [1]. For an autonomous underwater robot to be able to monitor and assist the human diver it is crucial that it has the ability to accurately detect and track the diver. While USBL measurements can provide relative underwater localization, its readings are very sparse and delayed due to acoustic wave propagation [2] and thus not suitable for a real time operation. The authors in [3] propose combining USBL measurements with forward looking multibeam sonar images to improve diver detection. Multibeam high frequency sonar devices provide almost real-time acoustic images with high precision and are commonly used in underwater positioning and object detection ([4], [5], [6]). Most sonar image processing techniques, including the ones mentioned include pattern recognition or blob detection and clustering and combine them with a variety of classification filtering methods like Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF), Gaussian Mixture Model (GMM), Hidden Markov Model (HMM) etc. [7].

Deep convolutional neural networks (CNNs) have long surpassed all known methods of large-scale image recognition and classification and are continuously keeping performance levels rising on all benchmark tests like ImageNet, PASCAL VOC and MS COCO [8]. However, the beforementioned benchmark datasets consist of optical images. The goals of this paper are to evaluate the performance of state-of-the-art object detection neural network architectures on forward looking sonar recordings and to evaluate the feasibility of using them in a real-time autonomous marine application. Several studies like [9] and [10] have been conducted in sonar-based object classification using machine learning approaches which provide the probability that an object is present in the image. In order to track the diver's trajectory and position using an AUV, a precise location inside the sonar field of view has to be determined. All of the proposed CNN architecture models were trained on the same training dataset and then evaluated on several validation datasets in order to measure their performance, computation requirements and execution speed.

The main motivation for conducting this type of research arises from the ONR-G "ADRIATIC – Advancing Diver-Robot Interaction Capabilities" project, one of whose main objectives is to develop new collaborative motion strategies between a human diver and an autonomous underwater vehicle. Diver's limited navigation capabilities and lack of communication with the surface when underwater introduce great risk in case an emergency situation happens. Technical life supporting equipment failure and external environmental disturbance risks are commonly minimized by diving in groups or at least in pair with a buddy. Using the buddy diving system not just in technical but in recreational dives as well is widely accepted not just as the standard operating procedure, but it has largely become the code of practice and part of governing legislation in counties around the world [11]. Within the ADRIATIC project the collaboration scheme between the diver and the AUV envisions the robot vehicle to take the place of a robotic diving buddy, the prime goal being observing the diver and determining his physiological parameters such as breathing, hearth and motion rate and to allow the detection of critical diver states.

This research is sponsored by the Croatian Science Foundation Multi-Year Project under G. A. number IP-2016-06-2082, named CroMarX and Office of Naval Research - NICOP research project ADRIATIC under grant number N00014-18-1-2011. The work of doctoral student Igor Kvasić has been supported in part by the "Young researchers' career development project – training of doctoral students" of the Croatian Science Foundation funded by the European Union from the European Social Fund.

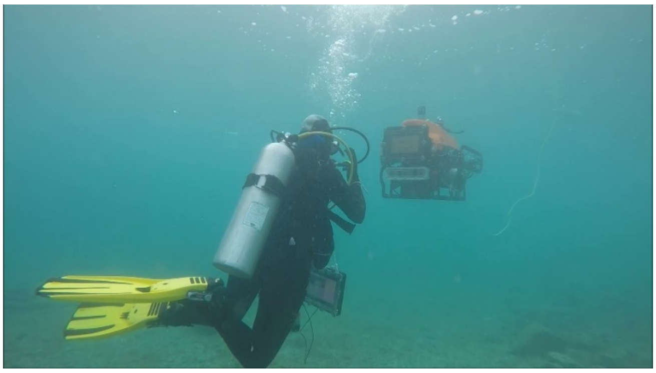

Fig. 1 Diver interacting with the BUDDY AUV, equipped with Soundmetrics Aris Explorer 3000 sonar

The diver-robot cooperation research focuses on three main scenarios. The first scenario makes the AUV act as an "observer" that monitors the diver at all times by keeping at safe distance and anticipating any difficulties the diver may experience. The second scenario uses the AUV as "slave" that assists the diver during some activities and can perform a limited set of actions, and the third "guide" option navigates the diver to a target or point of interest. The key in making all three scenarios possible lies in a robust method of detecting the diver's position and orientation relative to the AUV at all times. In addition to the diver-robot cooperation objective, the project explores novel human-robot communication methods and evaluates the feasibility of the proposed interaction solutions in real environment, which also requires a precise and reliable estimation of the diver's pose and location.

II. Sonar image processing

Using optical image cameras for underwater object detection tasks can present a great challenge with the increase in distance because thicker bodies of water often introduce a critical decrease in visibility. Due to this reason, in many underwater operations where perceptual object detection and recognition is required regardless of the water turbidity, imaging sonar has been widely accepted for providing reliable measurements [6]. Sonar can provide perceptual imaging of underwater environment in low or even zero visibility conditions. This is particularly critical in military and defense operations, search for evidence and search and rescue operations.

When it comes to sonar technologies, there are two types of sonar that can be used for object detection. Side-scan sonar (SSS) can provide long-range high-resolution imagery and it's suitable for performing detection in vast survey areas. On the other hand, multibeam forward-looking sonar (FLS) can acquire much higher detail levels but at shorter distances [13]. Because of these features they are often referenced as acoustic cameras because they, like a video camera, produce a two-dimensional image but using a very different geometrical principle. For a target like a human diver to be detected and the chosen detection principle the former type of sonar is much more suitable. FLS are constructed in a way they can emit a number of acoustic beams, each of whom covers a certain horizontal and vertical angle. Because of the different projection model than in an optical imaging camera, a sonar image often becomes hard to interpret and it usually does not come intuitively to a human operator to instantly define and recognize visual features in it. Additionally, a significant amount of noise is generated in the image due to the sensor's physical characteristics, so the image has to be preprocessed in order to become useable and to determine the target position.

The sonar data used to train deep neural networks used in this paper is acquired during the "CADDY - Cognitive Autonomous Diving Buddy" project validation trials during October 2015 and October 2016. The experimental setup consisted of the Soundmetrics ARIS Explorer 3000 sonar [14] mounted horizontally on a custom built autonomous underwater vehicle "BUDDY AUV". You can find out more about the concept of the vehicle and its characteristics in [15]. The sonar uses 128 horizontal beams, each covering a 0.25° angle which makes up a field of view of 30° horizontally and 15° in the vertical plane. It supports two operating modes; detection mode that ensonifies a 15-meter range at a frequency of 1.8 MHz and the identification mode with higher detail at ranges up to 5 meters operating at 3 MHz.

III. Convolutional Neural network object detectors

Computer vision is one of the fields that has been developing rapidly thanks to deep learning. These advances in computer vision are enabling brand new applications of this technology that just were not possible or feasible a couple years ago. Underwater robot vision is certainly one of the fields that could benefit from neural network research. Thanks to the global tendency of opening up the artificial neural network programming community more and more companies and individuals are encouraged to contribute together in open source libraries, constantly competing with each other and making the edge of this technology move forward in a very fast pace. Exactly for that reason it makes sense to test out the current state-of-the-art network architectures and topologies before proceeding into building your own or making modifications that meet your specific needs.

Building a deep neural network that performs object detection on a high-resolution image using standard fully-connected network layers would be infeasible since the number of parameters escalates very quickly when every pixel of every channel in the image gets multiplied by its own weight factor in the network layer matrix. That is part of the reason why the convolution operation become one of the fundamental building blocks of any modern object detection neural network architecture. That way features across the image can share filters with the same weights and still get detected anywhere inside the image. How convolution operations work on a simple edge detection problem is very well explained in [12]. Combining convolution layers with pooling, activation and fully connected classification layers make the neural network able to detect lines and curves in first shallower layers and more complex shapes or complete objects in deeper layers.

The objective of the network architectures taken into consideration in this paper is not just to detect if an object is present in the image or output the probability of its existence, but it is also responsible for drawing a bounding box around it and detecting its exact position within the image. This turns the classification problem into a detection with localization problem. Historically, object detection and localization started with the "sliding window" approach, where a window much smaller than the image size was cropped out of the image and passed sequentially to a neural network to make predictions. The process is repeated across the image to find objects, after which it is done all over again with larger window sizes. Naturally, this kind of approach resulted very computationally expensive. Some of the classic network examples from that era were LaNet-5 form 1995, AlexNet from 2012 and VGG-16 from 2015. The solution to that problem lies in a simple hack where the fully connected layers of the network are replaced with 1x1 convolution layers which at first does not sound particularly useful (a 1x1 matrix operation is just multiplying with the same number) but it comes very useful to shrink the number of layers and add some non-linearity to the model. Some famous networks that used that principle were Inception or GoogleNet. In 2015, the Residual Neural Network (ResNet) introduced a novel architecture with "skipping connections" which allowed to train bigger networks much faster and with lower complexity. Adding residual blocks does not hurt the network performance as it acts as a "same" convolution, but this "shortcut" allows the gradient to be directly backpropagated to the earlier layers of the network. Another ground-breaking idea that made the detection much more accurate and faster than the sliding window algorithm is YOLO (You Only Look Once). The idea is to do a segmentation of the original image into multiple grids implemented convolutionally, where each cell outputs a prediction vector and associates that prediction to an anchor box. The cell with the highest confidence score makes the prediction for an anchor box shape with the highest probability (Non-max Suppression). In this way the model does not process segments of the picture separately but processes the whole image at once which is a very powerful idea.

As the scope of this paper we chose to give an overview of five different CNN architectures and show how they perform on the same training and test data. The first network architecture we tested was the VGG-16 (or VGGNet) which was developed in 2014 by Zisserman and Simonyan [18] and it originally consisted of 16 layers in a very uniform structure made of 3x3 convolutional filters, pooling and fully connected layers. YOLO v2, also referenced as YOLO9000 is the second version of the YOLO architecture and was published by Redmon and Farhadi in 2017 [19] and it introduced major improvements to the original YOLO architecture to make it a better, faster and stronger object detection algorithm than the main rival at the time, Faster R-CNN. Some of the improvements it introduced are a higher resolution classifier, batch normalization, multi scale training etc. It featured 19 convolutional layers, 5 pooling layers and a Softmax classification layer. YOLO v2 was soon surpassed by its third version, YOLO v3, a much deeper network featuring 53 mainly 3x3 and 1x1 convolutional layers and a couple more incremental improvements. One of them was the use residual blocks or previously mentioned "shortcut connections", which allowed for faster training of the deeper network, but it also introduced the logistic classifier instead of the Softmax function, Feature Pyramid Networks (FPN) and more. When exploring further real-time operation options, the Tiny-YOLO v3 network was considered. Essentially, it is a lighter version of the YOLO v3 architecture limited to just 21 layers that uses downsampling in the pooling layers to reduce matrix dimensions in early layers. Taking everything into consideration, it is intended to be a tradeoff between performance and speed and more suited to a small to middle size dataset with faster execution on limited hardware resources. Finally, the most up-to-date model in the moment of writing this paper that was tested was the newest modification to the architecture and currently the highest scoring YOLO architecture, the YOLO v3-SPP. This configuration introduces a dense connection and Spatial Pyramid Pooling (SPP) principle to strengthen the feature extraction and alleviate the vanishing-gradient problem as well as concatenate multi-scale local region features. As a result, the network is able to learn features more comprehensively and it is currently achieving the highest mAP scores on PASCAL-VOC and UA-DETRAC datasets.

IV. Training Set Composition

As with any deep learning project, composing and preparing a good dataset is always the most important task. The more images with the target shown from different sides, relative sizes, angles of rotation, tilt, illumination etc., the more chance the network is going to have at detecting the object in different real-life conditions.

Sonar images used to train the networks in this paper were acquired by extracting frames from the Soundmetrics Aris Explorer 3000 video recordings. The frames come in a native resolution of 350x658 pixels and all the CNN input filter matrices are modified to fit as close as possible to that resolution in order to increase the network resolution. That should allow for detection of smaller objects, meaning there should be a higher possibility of detecting the diver further away in the sonar field of view. The base of the dataset consists of 2000 images for the training set and 200 (10%) images for the test set. The images are chosen from random non-sequential frames of different sonar recording to maximize the diversity of the training dataset and to prevent it from overfitting.

The images in the dataset are recorded in seawater during experimental trials in Biograd na Moru, Croatia, with the diver recorded by the AUV form different perspectives, in different body poses and orientations, at different distances from the vehicle. An example of how the sonar recordings of the diver look can be seen in Figure 2. Including all perspective combinations is crucial for the network to do inference properly and to adapt to various conditions it might encounter on validation trials. Although 2000 images might not seem a lot considering computer vision algorithms use tens of thousands of images for training, it is enough to give an analysis of the network performance and speed, especially considering the fact that we are training the network to detect only one class of objects. The dataset is further expanded using data augmentation techniques. Considering we did not post the requirement to distinguish left from right sides on the detected objects, random scaling, translations and x-axis mirroring is done to multiply the dataset and prevent it to overfit early, as well as randomly adjusting exposure, hue and saturation. It is also important to include negative samples that do not contain any of the objects we wish to detect in the dataset without any bounding box labels. Authors in [16] suggest using as many negative samples as there are images with positive labeled objects. All of the images from the training and the test dataset containing the diver in the sonar image were labeled manually using a standard Python labeling tool. The tool generates a text file for each of the images containing the label class number of the object in the bounding box, x and y coordinates of the bounding box center, relative to the left bottom corner, and the width and height of the bounding box, all scaled to numbers between 0 and 1 (1 being the size of the image in the given axis). Once prepared in this manner the labeled bounding box is considered to be the "ground truth" and the dataset is ready to start the training process.

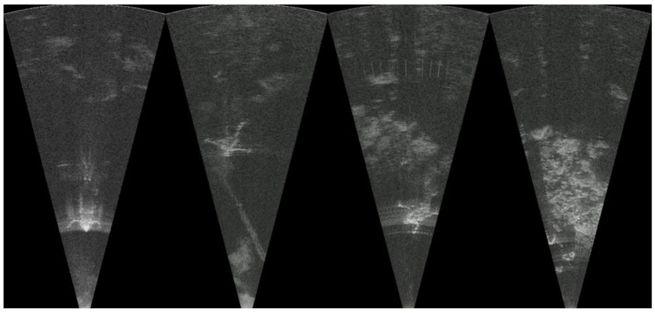

Fig. 2 Examples of sonar images acquired using Aris Explorer 3000 in seawater experimental trials.

Using widely available pre-trained model weights for starting the network training process resulted in the best convergence times and a faster learning process than any other option. Although the used model weights were trained on the COCO dataset, that does not present any similarities with the type of images we want to train on, the network adapted these weights much faster than when just using random or zero weights.

V. Training

During the training process the deep neural networks learn the parameters of the model, which for a CNN are the filter weight factor matrices with the corresponding bias factors. The training process tries to minimize the classification error on the test data and repeats the process iteratively on batches of images randomly chosen from the training dataset. The batch sizes used for training the networks in this paper was 64 images at once. All the object detection training was performed on a PC running the Nvidia CUDA toolkit, which allows parallel distribution of computing on two graphics processing units Nvidia GTX 1080 Ti. The training process is sped-up further using the OpenCL framework which enables the learning to be executed on the central processing unit as well, in our case and Intel Core i9 9900k with 8 cores and 16 threads running at frequencies up to 5 GHz.

The outputs of the CNN are very similar to the described annotation labels data we used to feed the neural network to learn, and it consists of coordinates of the center of the predicted bounding box, its width and height relative to the size of the image and the class the object belongs to. Along with the class prediction, the network also outputs the confidence score for that particular prediction, or the probability that the detected object belongs to a certain class for a given intersection over union threshold. One set of mentioned parameters are output for each of the objects found in the image.

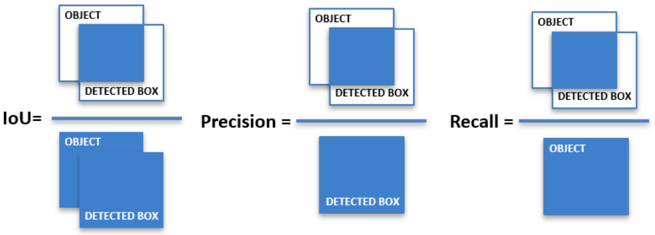

Fig. 3 Precision metrics definitions in object detection using bounding boxes – Intersection over Union, Precision and Recall.

Before we start analyzing the results, let's first explain the evaluation metrics used to compare different deep neural network architectures. Considering we are dealing with bounding boxes and object detection rather than just image classification, *precision* and *recall* terms are going to have slightly different meanings. In a classification problem precision would be defined as the ratio between true positive classifications over the sum of true positive and false positive classifications. As we are working with bounding boxes, it sounds reasonable to express precision as the ratio of the intersection surface between the labeled ground truth bounding box and the one the network actually detected and the surface of the detected box itself. Similarly, recall in a classification problem is defined as the ratio between true positive classifications over the sum of true positive and false negative classifications. In the bounding box surface analogies that would translate into the ratio between the intersection surface of the labeled and the detected bounding box and the ground truth bounding box surface. *Mean average precision (mAP)* is the default metric of a network precision in object detection and it is defined as the average value of 11 points on the precision-recall curve for each possible detection threshold averaged across all classes [17]. Another relevant evaluation metric is the *intersection over union (IoU)* value, which is defined as the area of overlap between the detected bounding box and the ground truth and area of their union. The IoU metric is particularly useful when adjusting the detection thresholds. All three of these metrics' definitions can be more easily understood when displayed graphically as in Figure 3.

For each experiment one type of CNN was trained using the same dataset and equivalent parameters adjusted to its specific architecture to get the best results. The learning curve varied broadly between different network models as expected, considering different depths and network parameters. However, all the networks were trained until the average loss reached a minimum, but early enough that the model does not start overfitting. You can find more useful guidelines on when exactly to stop the training process in [16]. A typical average loss function curve during training can be seen in Figure 4. When the average loss stops decreasing by any significant amount with further training steps, it is time to find the point that gives the highest mean average precision on the test dataset. The lowest average loss does not necessarily guarantee the best performance when doing validation on new data as the model could have started overfitting to the training data. Once the model has finished training, we chose the weights with the highest mAP and IoU and tested them on previously unseen sonar video recordings to evaluate their performance and model inference speed.

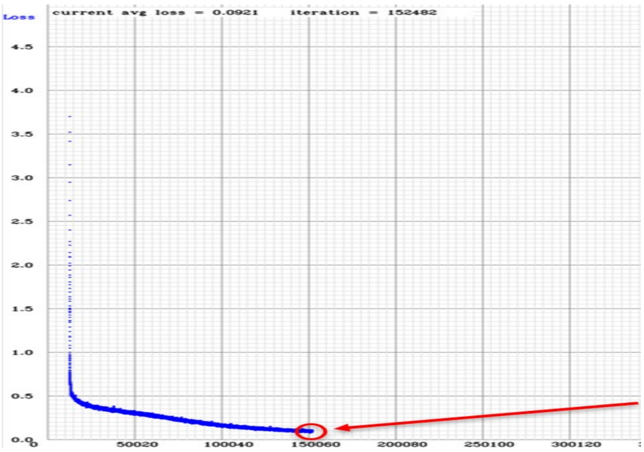

Fig. 4 Average loss curve across training iterations during Yolov2 model training. Indicated in the image is the "early stopping point", where we get the maximum mean average precision on the test dataset

VI. Results

As mentioned earlier in the paper, the goal of this analysis was to test state-of-the-art CNN architectures on sonar recordings to rank their performance but also evaluate the feasibility and robustness of using such approaches in an autonomous underwater application. For that reason, the parameters we are going to look at when ranking the networks, beside just precision and recall, are going to be the complexity of the network, execution frame rate (frames per second, FPS) on a given hardware setup and the number of floating-point operations that need to get computed in a second of inference.

The results are reported as mAP at the IoU threshold set to 0.5 and the detection confidence threshold of 0.25 as shown on Figure 5., meaning the performance measure counts the prediction as true positive if the intersection of the prediction and the ground truth bounding boxes is greater than 50% of their union. Because the performance metric places much more emphasis on the bounding box placement accuracy the mAP results are not extremely high (i.e. in a 90% to 100% region). Part of the reason to that is that it is often hard to determine the shape and exact borders of the bounding box even during dataset labeling. Flippers the diver is wearing, for example, can often be seen only vaguely (Figure 7) and we can see during inference the model is often struggling to predict the shape of the bounding box, but the diver body position gets predicted correctly nevertheless.

In the execution speed evaluation shown on Figure 6 we can see that all the CNN models are outclassed by Tiny YOLO v3, as expected considering the fact that model is designed to run fast on limited hardware resources. What is interesting to note is that the mAP score for that same network is on par with YOLO v3 and YOLO v3-SPP and performed significantly better than YOLO v2 and VGG-16. The VGG-16 architecture beside achieving a low mAP score is simply not feasible to use in a real-time operation. The frame rate of the other networks on the other hand, although running on powerful hardware, is well above the expected sonar readings rate (Soundmetrics Aris Explorer 3000, used in this experiment, has a frame rate of up to 15 FPS). A full list of the performance indicators can be found in Table 1. The table features the already mentioned mAP, FPS, precision and recall metrics as well as BFLOPS (Billions of Floating-Point Operations, sometimes referenced as Giga FLOPS) as the measure of complexity of the model.

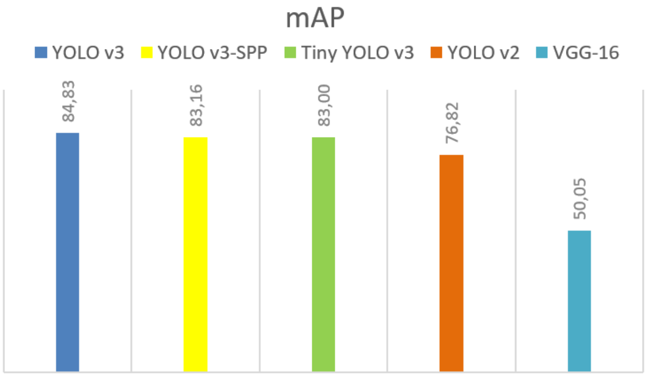

Fig. 5 CNN architectures ranked by the Mean Average Precision score on the same test dataset at IoU threshold set to 0.5 and detection threshold 0.25.

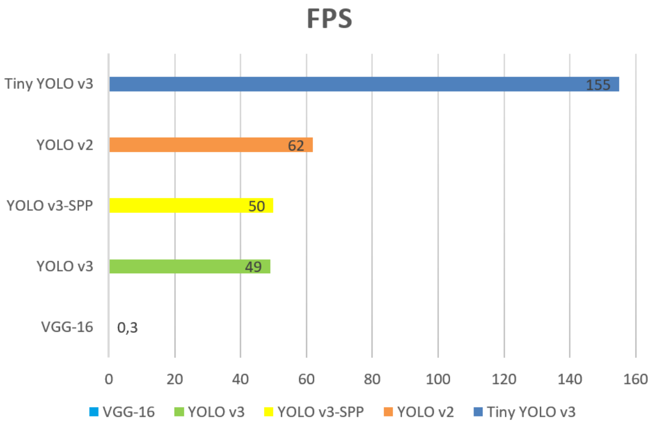

Fig. 6 Average frame rate results while performing inference on the same sonar video running on the hardware setup described in chapter V. CNN models are ranked from fastest to slowest.

TABLE I. CNN Performance Indicator Scores

Model	*Layers*	*mAP*	FPS	BFLOPS	Precision	Recall
VGG-16	16	50.05%	0.2-0.3	30.699	0.47	0.55
YOLOv2	26	76.82%	56-68	34.720	0.81	0.84
Tiny YOLOv3	21	83.00%	120-190	5.571	0.89	0.85
YOLOv3	53	84.83%	46-52	65.864	0.92	0.92
YOLOv3-SPP	53	83.16%	47-53	29.371	0.88	0.89

VII. Conclusion and Future Work

Overall, modern CNN object detector architectures have shown some astonishing results detecting even an unusual custom object like a diver within a sonar recording. It is still not completely clear or maybe even a bit counterintuitive how neural networks learn to recognize features as well as they do, a lot of the times performing that task better than a human would. It is, however, certainly something to exploit to our benefit and it is already making a huge impact in computer vision in warehouse autonomous robots, self-driving cars, drones and many other terrestrial and areal applications. Although perceptual and communication technologies might be very different underwater, there is no reason similar approaches cannot be used in underwater systems and technologies.

The network architectures proposed in this paper were originally not designed for doing inference on a single object or a single class of objects as we did in this test. They were designed to detect a large number of objects simultaneously, as i.e. vehicles and pedestrians on a busy street crossings, and to be trained in tens, hundreds or even thousands of different object categories (i.e. YOLO9000). The reason they were chosen to be tested in an underwater autonomous application is because they are incredibly fast in real time operation and they can be executed on limited hardware resources you would usually get in an AUV setup. Considering sonar recordings get updated in an order of magnitude of Hz or tens of Hz, we can see that some of the proposed architectures fit well above that frequency and show extraordinary detection precision results.

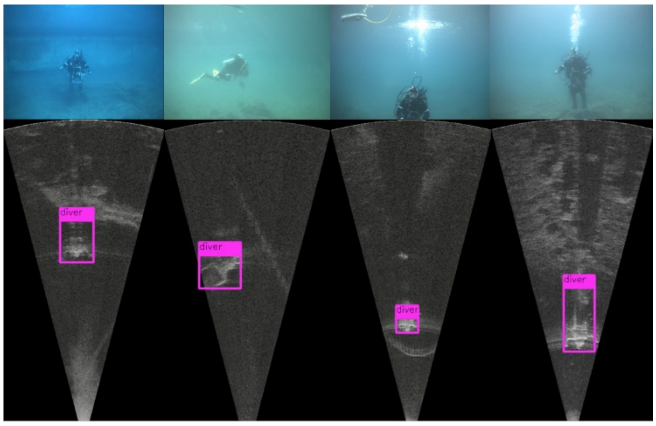

Fig. 7 Diver detection algorithms running on different sonar recordings with corresponding camera frames for reference.

Although sonar image recordings by definition do not provide any color information (color mapping might be added in post processing) we still used three channel RGB images to train the network models. Switching to mono-channel grayscale images for training could additionally speed up the training and inference process considering all the weight matrices get decreased by one dimension. The reason for not discarding the extra color layers is solely in demonstration and performance analysis purposes, considering grayscale trained models could not perform inference on video but just on single images. Future plans are to train models on one-channel images and perform inference sequentially on single images in a tracking filter which should additionally speed up the process, as well as decrease the on-board hardware requirements. Expanding the sonar database is one of the main goals for improving the detection and it is scheduled for the 2019 Breaking the Surface conference and workshop where preliminary sea trials are going to be done. Expanding the negative samples dataset of different seabed compositions could also improve the network performance. Other future work plans include expanding the learning database to multiple classes making the AUV able to detect multiple different objects of interest. Another field of interest for future research would include estimating the orientation of the diver which would prove very useful for the ADRIATIC project objectives.

References

[1] N. Mišković, M. Bibuli, A. Birk, M. Caccia, M. Egi, K. Grammer, A. Marroni, J. Neasham, A. Pascoal, A. Vasilijević, Z. Vukić, “CADDY—Cognitive Autonomous Diving Buddy: Two Years of Underwater Human-Robot Interaction”, Mar. Technol. Soc. J. 2016, 50, 54–66.

[2] M. Morgado, P. Oliveira, C. Silvestre, and J. F. Vasconcelos, “Embedded vehicle dynamics aiding for USBL/INS underwater navigation system,” IEEE Transactions on Control Systems Technology, vol. 22, no. 1, pp. 322–330, 2014.

[3] F. Mandić, I. Rendulić, N. Mišković and Đ. Nađ, “Underwater Object Tracking Using Sonar and USBL Measurements”, Journal of Sensors, vol. 2016, Article ID 8070286, 10 pages, 2016.

[4] T. Zhang, W. Zeng, L. Wan, and S. Ma, “Underwater target tracking based on Gaussian particle filter in looking forward sonar images,” Journal of Computational Information Systems, vol. 6, no. 14, pp. 4801–4810, 2010.

[5] D. W. Krout, W. Kooiman, G. Okopal and E. Hanusa, “Object tracking with imaging sonar,” in Proceedings of the 15th International Conference on Information Fusion (FUSION ’12), pp. 2400–2405, IEEE, Singapore, September 2012.

[6] E. Galceran, V. Djapić, M. Carreras and D. P.Williams, “A realtime underwater object detection algorithm for multi-beam forward looking sonar,” IFAC Proceedings, vol. 45, pp. 306–311, 2012.

[7] K. J. DeMarco, M. E. West and A. M. Howard, “Sonar-based detection and tracking of a diver for underwater human-robot interaction scenarios,” in Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC ’13), pp. 2378–2383, IEEE, Manchester, UK, October 2013.

[8] A. Krizhevsky, I. Sutskever and G. E. Hinton, “Imagenet classification with deep convolutional neural networks” In NIPS, 2012.

[9] M. Valdenegro-Toro, “Object recognition in forward-looking sonar images with Convolutional Neural Networks”, OCEANS 2016 MTS/IEEE Monterey, 1-6

[10] D. Williams, "Demystifying Deep Convolutional Neural Networks for Sonar Image Classification”, in Underwater Acoustics Conference, Skiathos, Greece, 2017.

[11] B. Halstead, "Line dancing and the buddy system: 52-54". South Pacific Underwater Medicine Society Journal. 30 (1). ISSN 0813-1988. OCLC 16986801., 2000.

[12] D. Marr, E. Hildreth, “Theory of edge detection”, Proc. of Royal Society Landon, B(207): 187-217, 1980.

[13] Sound Metrics Corp., “About Sonar Imaging”, [Online] http://www.soundmetrics.com/About-Sonar-Imaging, 2019.

[14] Sound Metrics Corp., “ARIS Explorer 3000”, http://www.soundmetrics.com/ products/aris-sonars/aris-explorer-3000, 2019.

[15] N. Stilinović, Đ. Nađ and N. Mišković, "AUV for diver assistance and safety - Design and implementation," OCEANS 2015 - Genova, Genoa, 2015, pp. 1-4.

[16] “Yolo-v3 and Yolo-v2 for Windows and Linux”, https://github.com/AlexeyAB/darknet, 2019.

[17] M. Everingham, M. A. Eslami, S, Van Gool, L., K. I. Williams, Christopher & Winn, John & Zisserman, Andrew, “The Pascal Visual Object Classes Challenge: A Retrospective.” International Journal of Computer Vision. 111. 10.1007/s11263-014-0733-5., 2014.

[18] K. Simonyan, A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition”. arXiv 1409.1556., 2014.

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