Leveraging Seabed-Context Information for Improved Underwater Target Classification Using Synthetic Aperture Sonar

Thibaud Berthomier[1], Bart Gips[2], David P. Williams[3], and Thomas Furfaro[1]

[1]Science and Technology Organisation - Centre for Maritime Research and Experimentation, Viale S. Bartolomeo, 400, 19126 La Spezia, Italy
[2]Machine2Learn, Stroombaan 4, 1181 VX Amstelveen, The Netherlands
[3]Applied Research Laboratory – Pennsylvania State University, State College, PA, USA

Corresponding author: Thibaud.Berthomier@cmre.nato.int.

Abstract: *Recent advances in deep learning have enabled accurate and efficient classification of underwater targets captured in synthetic aperture sonar. The success of these models relies on the availability of large datasets with highly textured and rich images from a variety of environments. However, performance decreases in complex seabeds, where targets are more difficult to detect and false alarms are more common – for example, seagrass drastically increases the difficulty of finding a target, and sand ripples increase difficulty when insonified from an angle orthogonal to the ripples. Moreover, the amount of available training data is often limited in these challenging environments compared to smooth seafloors. High-resolution images enable relatively straightforward visual recognition of seafloor type: smooth, sand ripples, vegetation, clutter,* etc.*In previous work, we developed an automatic seabed characterization algorithm based on Gaussian processes. In this work, we aim to leverage this knowledge in order to improve target classification performance. To this end, we introduce a new context-dependent classification algorithm to address and exploit the environment. Based on our previous ATR framework, we implemented and trained convolutional neural networks (CNNs) employing two strategies: by injecting the seabed information directly into the decision-maker (*i.e. *the CNN) and by specializing the CNNs (*i.e. *fine-tuning) for each class of seabed, with the decision being made using the outputs of the CNNs and the seabed prediction. The effectiveness of our approach will be demonstrated using real data collected during at-sea experiments.*

Keywords: *convolutional neural network, gaussian process, automatic target recognition, seabed characterization, synthetic aperture sonar.*

1. INTRODUCTION

The success of Convolutional Neural Networks (CNNs) in remote sensing can be attributed to their ability to detect spatial patterns in images and capture contextual information. However, the effective employment of CNNs is not immune to challenges such as class imbalance, data distribution discrepancies between training and test datasets, and strong dependence to the environment [1, 2, 3]. If CNNs demonstrate exciting and, in some cases, human-competitive results [4], the ability of discriminating mines from clutter in sonar imagery depends greatly on the complexity of the environment. The seabed composition has a large impact on the perception of the mine by the sonar [3, 5, 6, 7].

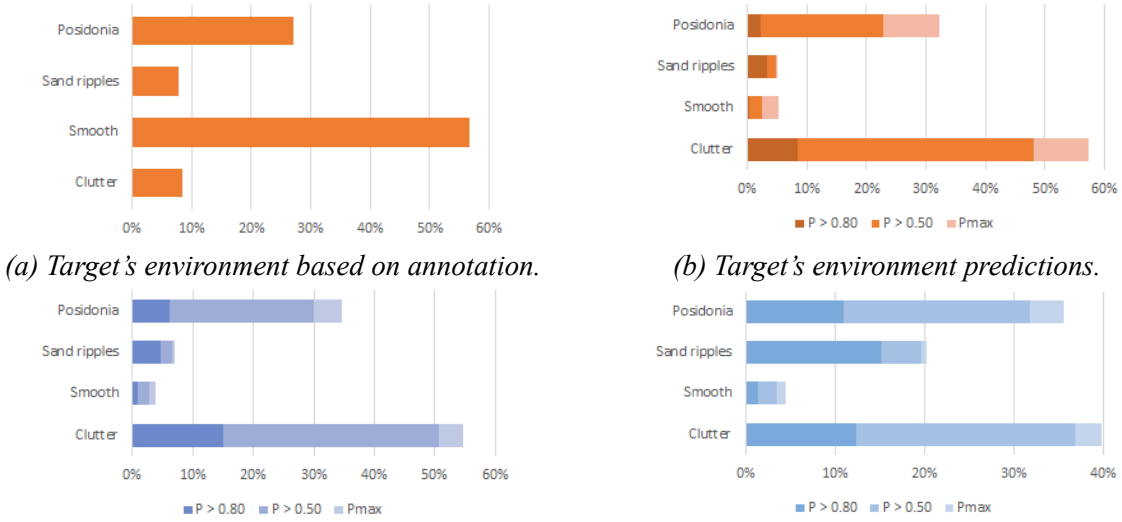
In previous work, we implemented efficient, small CNNs for underwater target classification [4, 8, 9], but also a texture-based seafloor characterization algorithm [6]; the purpose of this work is to take into account the environmental information nearby the targets to enhance the classification accuracy of the CNNs (in SAS images). The seafloor is characterized according to four types of textures – smooth, with sand ripples, vegetation or clutter – using a Gaussian Process classification model (*cf.* Section 2). To exploit these outputs, four new CNNs are created by fine-tuning the original CNN on each class of seabed (Sections 3 and 4) and combined with the objective of improving the classification performance (*cf.* Section 5).

2. SEABED CHARACTERIZATION

This section describes how the bottom type characterizer was applied for our purpose. The algorithm, based on Gaussian processes (GPs), is fully described in [6]. The GP classification (GPC) model predicts the seabed type associated to sub-areas of an image, and provides an uncertainty estimate on this prediction. The used model was specially trained for this work (following the same procedure) to handle the small sonar images (called *mugshots*) that are given as input to the underwater target classification algorithm.

The GPC model was applied to our training and test datasets, which are detections from the Mondrian detector [10]. It represents more than 600 000 mugshots (335 $\times$ 335 pixels, 5.025 $\times$ 5.025 meters) including a few thousands views of targets (*i.e.* highly unbalanced datasets). For each mugshot, a larger image (535 $\times$ 535 pixels, 8.025 $\times$ 8.025 meters) was extracted from the full SAS image and segmented with the GPC model. Because the targets themselves obscure the sea bottom, we exclude the target highlight and shadow regions and took the average of the remaining values to compute the probability that a target (or false alarm) is surrounded by each class of seabed.

Fig. 1 provides the results of the automatic characterization of the seabed that is surrounding the detections. As you can observe, the distribution of the images (both target and non-target cases) is quite unbalanced according to the bottom type class prediction. The bottom type classes of the images are mainly associated to the classes Posidonia (a type of marine vegetation) and clutter. These two classes represent a minority of the full database but these complex bottom types, as well as sand ripples, often cause false alarms. This probably explains why there are few cases associated to smooth seabeds.



(a) Target's environment based on annotation.

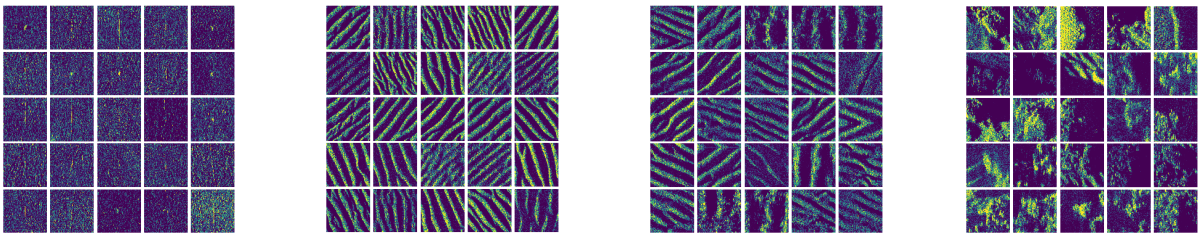
(b) Target's environment predictions.

(c) Predictions of all non-target cases.

(d) Non-target cases of the evaluation subset.

Figure 1: Training and test data associated to each seabed class using the annotation tool or the GPC model for three decision cases: probability P higher than 0.80, 0.50 and all other seabed scores.

By comparing the seabed predictions (see Fig. 1a) with the annotations (see Fig. 1b), we notice that many targets surrounded by a smooth seabed were associated to the clutter class. It means that the targets' highlight and shadow impact the bottom type characterization and that the model classifies as clutter a smooth seabed when a target is laying. For the same reason, many non-target mugshots could be associated to the clutter class, even though they are surrounded by a smooth seabed. For future work, we should probably redesign the subarea selection used to define the bottom type features. Fig. 2 gives examples for each class of seabed. Based on this small sample, we can assume that many cases of Posidonia are actually sand ripples, probably with a large wavelength. This is not surprising since the Posidonia class include images where the Posidonia is laying on sand ripples.



(a) Smooth. *(b) Sand ripples.* *(c) Posidonia.* *(d) Clutter.*

Figure 2: Training images associated to seabed classes with a probability higher than 0.90.

A small subset of 56 000 images, all targets included, (called *evaluation dataset*) was selected to evaluate quickly the proposed CNN architectures before training them on the full dataset, which requires more computational time. This subset, also considered in previous work, is still unbalanced according to the seabed class (see Fig. 1c). In previous work [4, 9], we demonstrated that this amount of data is sufficient for fine-tuning and even to train fully a CNN. Indeed, if the number of free parameters is comparable to the size of the datasets, the CNN are able to generalize well enough for the binary classification task using SAS imagery.

3. UNDERWATER TARGET CLASSIFIER

This section describes the convolutional neural network (CNN) that we used to perform underwater target classification. The CNN input is an output of a detection algorithm and the output is the probability of belonging to each class under consideration (*e.g.* target and non-target). Different architectures were proposed in [4], but only one of them (called CNN B) is considered. Its architecture is given by Tab. 1. In this model with 1509 free parameters, the feature-extraction stage consists of four blocks, each containing a convolutional layer, a rectified linear unit activation (ReLU) and an average pooling layer. The classification stage is a dense (fully-connected) layer followed by a final sigmoid activation function to estimate the probability that a mugshot belongs to the target class. The input is a 267×267 pixel SAS image mugshot, where pixels span 15 mm in each dimension. The numbers of convolutional filters and pooling operators were selected such that the final dense layer always contains only four nodes. That is, the network is forced to predict the class label from only four features.

Table 1: Architecture of the pre-trained classification CNN used for the experiments.

Block	Layer	Filters	Kernel size	Output size
	Input	-	-	$267 \times 267 \times 1$
1	Convolution	4	$8 \times 8 \times 1$	$260 \times 260 \times 4$
	ReLU	-	-	$260 \times 260 \times 4$
	Pooling	-	$4 \times 4 \times 1$	$64 \times 64 \times 4$
2	Convolution	4	$6 \times 6 \times 4$	$60 \times 60 \times 4$
	ReLU	-	-	$60 \times 60 \times 4$
	Pooling	-	$4 \times 4 \times 1$	$15 \times 15 \times 4$
3	Convolution	4	$4 \times 4 \times 4$	$12 \times 12 \times 4$
	ReLU	-	-	$12 \times 12 \times 4$
	Pooling	-	$2 \times 2 \times 1$	$6 \times 6 \times 4$
4	Convolution	4	$5 \times 5 \times 4$	$2 \times 2 \times 4$
	ReLU	-	-	$2 \times 2 \times 4$
	Pooling	-	$2 \times 2 \times 1$	$1 \times 1 \times 4$
5	Flatten	-	-	4×1
	Dense	1	1×4	1
	Sigmoid	-	-	1

Training a CNN means learning the filters, and the associated bias terms, of the convolutional and the dense layers. This process was performed with the *RMSprop* optimizer – a technique that modifies the standard gradient descent algorithm of the training process [11] – combined with the binary-cross-entropy loss function. A batch contains 32 images of each class (so 64 images in total) to address the natural imbalance in the numbers of target and non-target mugshots. Data augmentation is also applied on each chosen mugshot: along-track reflections, and small translations in the along-track and range dimensions. The CNN was trained for 100 epochs of 1000 batches, with a learning rate of 0.001, which took approximately one day using one graphics processing unit (GPU).

The final receiver operating characteristic (ROC) curve and area under the ROC curve (AUC) were computed on the evaluation datasets for each predicted types of seabed. Theses results are also presented in the first row of Tab. 2. In comparison with the results obtained on all types

(*i.e.* on the full test dataset), the best results were obtained on smooth seabeds, where the AUC is 2.16% higher. On the types Posidonia and clutter, the AUC is similar then on all seabeds: $+0.31\%$ and -0.51%, respectively. The potential for improvement is the biggest in sand ripples where the AUC is 1.54% lower than on all seabeds. A deeper analysis of the results shows that the performance on smooth seabed is very high: false alarms are mainly due to clutter objects with mine shape while misclassification are due to poor image quality. In conclusion, and based on the bottom type characterizer outputs, there is room to improve the classifier's level of performance, in particular on sand ripples that seem to affect the most his accuracy.

4. SPECIALIZATION OF THE CLASSIFIER PER SEABED

In this approach, we aim to specialize the CNN for each class of seabed and build the final prediction based on the outputs of the seabed classification and the four new CNNs. Training the CNN for each seabed should fine-tune the first layers of the CNN to adapt to the environment. The fusion of the predictions of the four specialized CNNs using the seabed classification scores should improve the model accuracy, especially for complex seabeds where the false alarm rate is high. For the experiment, we consider that an image contains a type of seabed if the associated probability was higher than 0.50. As the number of target images are relatively low, all of them were used for the training whatever the predicted seabed class. We first trained the CNNs on the evaluation subset (about 56 000 images) and then on the full datasets (900 000 images). The CNNs were fine-tuned with 50 epochs on the evaluation subset and 150 epochs on the full dataset. Each epoch had 83 batches of 64 images, which represents the number of batches to see all the targets if they were not chosen randomly. The learning rate was fixed at 0.0001.

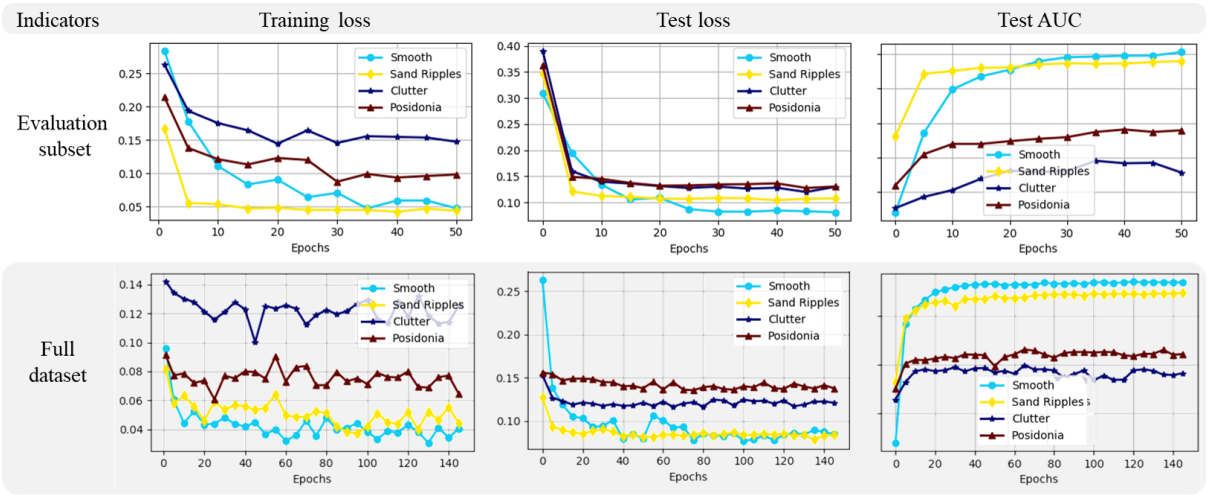


Figure 3: Training loss, test loss and AUC obtained while fine-tuning the specialized CNNs on the evaluation subset and the full dataset.

The outcomes of the training phase are summarized by Fig. 3. Three indicators are displayed: the loss (*i.e.* the binary cross entropy) on the training dataset, the loss and the AUC on the test dataset. In both cases, we can observe that the loss is decreasing on both the training and the test datasets, which means that the specialization of the CNNs is working. The gain is small on the AUC (less than 3%) but it can be substantial for the reduction of false alarms as the classes are unbalanced. The use of the full dataset provides similar results: they are slightly better on smooth seabeds and sand ripples where the amount of training data are limited (*cf.* Section 2),

but slightly worse on clutter and Posidonia. The final models were chosen by minimizing the AUC on the test set for each seabed.

5. EXPERIMENTAL RESULTS

Eight new CNNs were trained on specific environments based on the outputs of the bottom type characterizer algorithm. Tab. 2 provides the gain in AUC compared to the initial classifier (first row) of each new CNN individually (from second to fifth row) and of the CNNs working together (last two rows). For each seabed class, we selected the CNN with best AUC on that class between the CNN trained on the evaluation subset and the one trained on all the training data. The ROC curves, and the AUC, were computed on the evaluation dataset for each bottom type class. Without environmental awareness, the initial classifier outperforms the new CNNs specialized on smooth and clutter classes but not the ones specialized on sand ripples and Posidonia, which are given a slightly better AUC. However, each CNN specialized on complex seabeds outperforms the initial classifier if the image contains the bottom type for which it was specialized. An overfitting due to the small amount of images classified as smooth seabed could explain why the specialized network trained on that seabed does not perform as expected.

The idea is now to combine these new CNNs together using the local environment features. To do so, we created a new classifier based on the four specialized CNNs that multiplies the probability of the image to belong to each seabed class with the output of the CNN trained on that seabed. The final score s_I of an image I will be:

$$s(I) = \sum_{e \in [smooth,\ sand\ ripples,\ posidonia,\ clutter]} s_e(I) p_e(I) \tag{1}$$

where s_e is the score of the CNN trained on data associated to the seabed class e and p_e is the probability that I contains that type of seabed e. As can be seen in Tab. 2, this solution (called *Ensemble* Σ) didn't improve the model accuracy but decreased it (-2.37% on AUC). One reason could be that the training images were associated to a seabed only if the probability to belong to that seabed class is higher than 0.5. To fix this issue, we proposed a second solution (called *Ref &* Σ) that takes the score of the specialized CNNs only if one of the seabed probabilities is higher than 0.5, else it takes the score of the initial classifier. Doing so, we improved the classification performance by 0.93% in terms of AUC. At the customary 0.5 decision threshold, the false alarm rate decreases by 0.1% (*i.e.* remains stable) and the probability of detection increased by 2.4% compared to the reference CNN.

Furthermore, we observe that the CNN trained on smooth environment has lower performance than the reference one, so we replaced the specialized CNN by the reference one on smooth environment (called Σ^*). Compared to the reference CNN, this change leads to the same AUC as using the smooth seabed CNN, a drop of 0.1% on the false alarm rate and a gain of 2.1% on the probability of classification at the 0.5 decision threshold. Similarly, as the CNN trained on Posidonia has higher performance than the reference CNN in overall, we replaced the reference CNN used when the seabed probabilities is lower than 0.5 by the CNN trained on Posidonia (called *Pos.* & Σ^*). Doing so, we achieve a gain on AUC of 1.03%, the same false alarm rate and an increase of 3.3% on the probability of classification with a threshold of 0.5.

Table 2: AUC gain computed on all the test dataset: images are associated to a seabed class when the seabed characterizer probability is higher than 0.5 *for that class. At the first row, the gain is relative to the AUC on all seabeds. From the second row, the gain is relative to the AUC computed on each seabed class.* Σ^* *indicates that the reference CNN replaced the smooth CNN on smooth seabeds.*

CNN(s)	All seabeds	Smooth	Sand ripples	Posidonia	Clutter
Reference	-	$+2.16$ %	-1.54 %	$+0.31$ %	-0.51 %
Smooth	-5.25%	$\mathbf{-1.51}$ **%**	-10.66 %	-5.13 %	-4.55 %
Sand ripples	$+0.10$ %	-1.11 %	$\mathbf{+3.45}$ **%**	$+0.82$ %	-1.76 %
Posidonia	$+0.31$ %	-0.50%	$+2.40$ %	$\mathbf{+0.92}$**%**	-1.14%
Clutter	-0.62 %	-0.10 %	-5.43 %	-0.82 %	$\mathbf{+0.62}$ **%**
Ensemble Σ	-2.37 %	-1.11 %	$+3.45$ %	$+1.23$ %	$+0.72$ %
Ref. & Σ	$+0.93$ %	-1.11 %	$+3.45$ %	$+1.23$ %	$+0.72$ %
Σ^*	-2.47 %	$+0.10$ %	$+3.45$ %	$+1.13$ %	$+0.72$ %
Ref. & Σ^*	$+0.93$ %	$+0.10$ %	$+3.45$ %	$+1.13$ %	$+0.72$ %
Pos. & Σ^*	$\mathbf{+1.03}$ **%**	$+0.10$ %	$+3.45$ %	$+1.13$ %	$+0.72$ %

6. CONCLUSION AND FUTURE WORK

In this work, we aimed to improve the performance of underwater target classification algorithms by leveraging the knowledge of the surrounding environment. A new classification framework that takes into account environmental information has been proposed. We applied texture-based seafloor characterizer on an 8 by 8 meters image centered on the contact (output of the detector) to estimate the local environment and used the extracted features to predict if the contact is a target (or not). The sea bottom characterizer algorithm segmented all the training and testing datasets (more than 600 000 ”mugshots”).

To exploit the outputs of the bottom type characterizer, we tuned to the first layers of the CNNs by training them on each specific seabed. We generated four specialized CNNs that, combined together based on the bottom type characterizer predictions, can outperform the reference CNN. However, to observe this gain of performance, the reference CNN (or the one trained on Posidonia) needs to be used when the seabed estimation is uncertain. It should not be excluded that we reached a plateau of performance on our datasets, which would explain the marginal gain.

The overall results of this new ATR framework that considers the environment are promising and could probably be improved by some small changes at different stages of the process. In particular, the estimation of the local environment could be more robust in order to limit the impact of the object echo and shadow in the seabed prediction. This could lead to a better seabed estimation but also to a more balanced training dataset. On the same idea, we could add images that are not linked to a detection (the training data are outputs from the Mondrian detector) for the bottom types that are under-represented. The way that the four specialized CNNs are combined could also be optimized. Finally, we could inject seabed features given by the GP model directly inside the CNN (*e.g.* in the dense layer) as we have done in previous work where we estimated the image quality.

7. ACKNOWLEDGEMENTS

This work was supported by the NATO STO Centre for Maritime Research and Experimentation (CMRE) with funding provided by the NATO Allied Command Transformation (ACT).

REFERENCES

[1] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, **106**, 249–259, **2018**.

[2] I. Valova, C. Harris, T. Mai, and N. Gueorguieva, "Optimization of Convolutional Neural Networks for Imbalanced Set Classification," *Procedia Computer Science*, **176**, 660–669, **2020**.

[3] D. P. Williams and E. Fakiris, "Exploiting Environmental Information for Improved Underwater Target Classification in Sonar Imagery," *IEEE Transactions on Geoscience and Remote Sensing*, **52**, 6284–6297, **2014**.

[4] 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**, 236–260, **2021**.

[5] L. Picard, A. Baussard, G. Le Chenadec, and I. Quidu, "Seafloor characterization for ATR applications using the monogenic signal and the intrinsic dimensionality," in *OCEANS 2016 MTS/IEEE Monterey*, 1–5, **2016**.

[6] B. Gips, "Texture-Based Seafloor Characterization Using Gaussian Process Classification," *IEEE Journal of Oceanic Engineering*, **47**, 1058–1068, **2022**.

[7] C. Delblond, I. Quidu, L. Picard, G. Le Chenadec, and A. Baussard, "Monogenic signal study for seabed classification," in *International Conference on Underwater Acoustics*, (Southampton, France), **2022**.

[8] T. Berthomier, D. P. Williams, and S. Dugelay, "Target Localization in Synthetic Aperture Sonar Imagery using Convolutional Neural Networks," in *OCEANS 2019 MTS/IEEE SEATTLE*, 1–9, **2019**.

[9] T. Berthomier, D. P. Williams, B. d'Alès, and S. Dugelay, "Exploiting Auxiliary Information for Improved Underwater Target Classification with Convolutional Neural Networks," in *Global Oceans 2020: Singapore – U.S. Gulf Coast*, 1–10, **2020**.

[10] D. P. Williams, "The Mondrian Detection Algorithm for Sonar Imagery," *IEEE Transactions on Geoscience and Remote Sensing*, **56**, 1091–1102, **2018**.

[11] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning*, **4**, 26–31, **2012**.