UTILITY OF MODELING AND SIMULATION IN DATA-STARVED SCENARIOS FOR UNDERWATER MACHINE LEARNING APPLICATIONS

JD Park The Pennsylvania State University, University Park, PA, USA
DP Williams The Pennsylvania State University, University Park, PA, USA
J Philtron The Pennsylvania State University, University Park, PA, USA
SF Johnson The Pennsylvania State University, University Park, PA, USA
DC Brown The Pennsylvania State University, University Park, PA, USA

1 INTRODUCTION

For underwater remote sensing applications, modeling and simulation is increasingly being used as a tool for sonar system design, performance prediction, algorithm development, and most recently, training and testing machine learning (ML) algorithms dedicated to automated target recognition (ATR). The emergence of deep learning neural networks has created a new demand for modeling and simulation, due to the requirement for large amounts of training data. Although optical image datasets exceeding one million samples are available, the same is not true for sonar image data due to the practical considerations of operating an oceangoing sensor and installing object fields underwater. This effort analyzes the utility of augmenting ATR training data sets with simulated data.

A challenge of using simulated data is that the data does not naturally contain the level of variability observed in the real world. For example, without specific effort to add realism, simulated datasets often lack the noise, interference, distortions, and position uncertainty found in real-world data. As such, overly “clean” simulated data may have lower utility for training ATR algorithms.

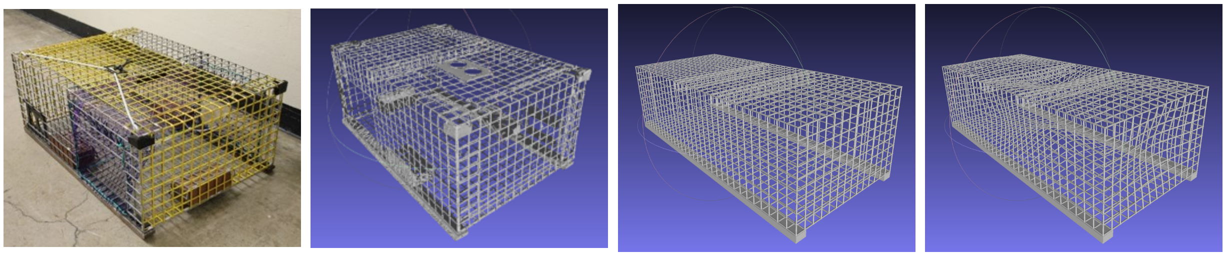
In this work, various elements of realism were incorporated when generating simulated acoustic data and reconstructing synthetic aperture sonar (SAS) images. The utility of simulated SAS imagery was quantified through the lens of ML in two ways. First, convolutional neural networks (CNNs) were trained to discriminate simulated and measured field data. It was observed that the ability of a CNN to discriminate these two sources of data decreases as the level of realism in simulated data increases. Second, a set of CNN-based ATRs were trained using different mixture ratios of simulated data and field data before testing on unseen field data. Results showed that the utility of simulated data is highest in data-starved scenarios, when relatively little measured data is available for training.

2 DATA MODELING

This section describes the data modeling approaches for developing realistic simulated data and validating the maturity of the simulated data for supplementing ATR training. The overall approach is to build models for the sensor, environment, and threat (SET), described in the Navy's vision for underwater mine countermeasure research and development[1].

2.1 Man-Made Object Modeling

Lobster traps were chosen for this study due to their abundance in measured data and their sufficient complexity that allows for control in modeling fidelity. A photo of one of the lobster traps is shown in Figure 1 (a). Multiple methods were used to create object models containing a 3D mesh of triangular facets. First, a direct laser-scan was conducted to automatically calculate a point-based mesh. This object model is realistic and contains the full realism of the trap, but the laser scanning process was very labor-intensive and required a significant amount of post-processing to remove spurious points. Secondly, computer-aided design (CAD) models were developed. The CAD model is simple, and lacks many of the detailed components. However, it allows for more control and variability in the final design. One of the important factors within our control with CAD models is the ability to add realism to the model by distorting the lattice pattern with "dents." This is discussed further in Section 2.5. The CAD model approach was used for simulation for its ease of use, the flexibility to add realism, and it provided sufficient level of details required to simulate acoustic time series data. Figure 1 shows a lobster trap photo and example object meshes.



(a) Lobster trap photo *(b) Laser scanned mesh* *(c) CAD model* *(d) Deformed CAD model*

Figure 1: (a) Photograph of one of the Lobster traps modeled and used in the acoustic simulations, (b) mesh generated from laser scan data, (c) CAD lobster trap models in an pristine, and (d) deformed state. Deformation is exaggerated for the purposes of visualization.

2.2 Environmental Modeling

In addition to the acoustic scattering off of objects, scattering from the sea floor is required for the full scene simulation. Realistic sea floor scattering modeling and simulation that generates raw acoustic time series data suitable for SAS processing is challenging. Power law roughness texture has been studied in synthetic aperture radar and sonar, which was applied to model sea floor texture. Synthetic rippled sea floor texture has also been developed[2] and used for this effort. Boundaries between different textures have been manually designed to create several scenes, in which objects were placed at random ranges and poses.

2.3 Acoustic Time Series Simulation

The Point-based Sonar Signal Model (PoSSM)[3,4] was used for generating simulated acoustic time series data suitable for synthetic aperture sonar image reconstruction. This simulation tool can be configured to generate raw time series data of scattered sound from the seafloor and objects by emulating the active sonar system traveling in any survey path, in a specified environment and object scene. Kirchoff scattering from the mesh triangular facets is used to calculate the scattered sound from objects[5]. For UUV-based linear SAS survey scenarios, the sensor travels nominally in a straight line. PoSSM generates data in the same format as data collected using field survey systems.

2.4 Signal Processing and Multiple Representations

In a typical remote sensing survey, the raw acoustic data is processed to generate imagery data products used for post-mission analysis (PMA) or other scientific analyses. ASASIN (Advanced Synthetic Aperture Sonar Imaging eNgine)[6] is used to generate SAS imagery from measured and simulated data. ASASIN is a time-domain back-projection image reconstruction software that utilizes a GPU for highly parallelized signal processing. The SAS imagery can be further processed to generate alternative data products such as the wavenumber domain representation, or k-space[7]. The k-spae representation has been shown to help improve the classification performance, and was useful for validating the fidelity of simulated objects.

SAS imagery is the primary source of information used during PMA, either by human operators or by automated algorithms. High resolution imagery achieved by SAS processing allows for reliable classification, which simulated data can also provide. However, the utility of simulated data for ATR training is dependent on the physical fidelity of the synthetic objects and sea floor depicted in the data. Simulation choices such as how object shadows are calculated can exist on a continuum of fidelity, and ATRs have sufficient capacity to exploit such detailed features.

2.5 Adding Realism to Simulated Data

An ideal, “clean” simulation environment configuration lacks many elements of realism. Without the addition of realism into the simulated data, the data will appear too crisp, blatantly contrived, and are easily distinguishable from field-measured data. In this study, several elements of realism were added, including ambient noise, mismatched bulk speed of sound, uncompensated motion of the sensor, and object imperfections. Ground truth was not available for each of these factors, physics-based engineering judgment and statistical characterization of existing measured data guided the implementation choices.

Deforming the lobster trap wire structure was one of the factors that had a significant impact on the simulation data fidelity. Lobster traps deployed in the ocean have been subjected to repeated use in harsh and dynamic conditions. The impact of deforming the wire structure was noticeable in the SAS image data, and especially obvious in the k-space representation. Deformations were induced by applying a maximum of 2 cm displacements to the lattice in several locations according to solid-model physics within ANSYS Mechanical software.

Figure 2 shows examples of image and k-space representations of SAS data chips containing a lobster trap on the seafloor. The rectangular lobster trap outline shown in the image is portrayed as angled highlights in the spectral domain, with the angle of the highlight corresponding to the trap’s physical orientation. When the trap is undeformed, these highlights occur in relative isolation (left subplot). However, for the deformed trap and in the field measured data these highlights are smeared and distorted (right subplot). The patterns shown in the k-space representation are indicative of typical lobster trap field measured data. The fact that we can represent key features of the lobster trap in k-space increases confidence in the physical realism of the simulation.

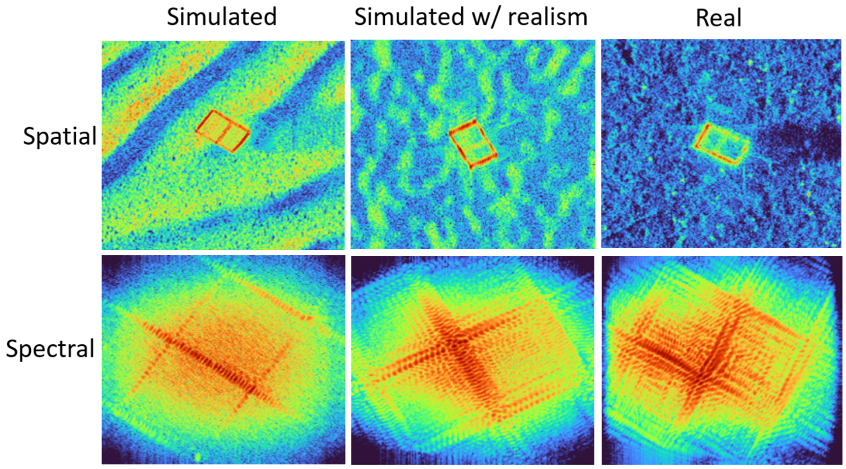


Figure 2: Example SAS image (upper) and k-space (lower) data representations. From left to right, a pristine simulated, deformed simulated, and field deployed lobster trap is shown.

3 ATR EXPERIMENTS

Two major studies were undertaken. The first study sought to assess the fidelity of simulated data to measured data. The second study sought to examine the impact of using simulated data to supplement measured data for classification. Both studies leveraged measured data collected by a high-frequency synthetic aperture sonar (SAS) sensor and simulated data was produced by the PoSSM tool.

3.1 Fidelity of Simulated Data

3.1.1 Data

The objective of this experiment was to attempt to discriminate measured data from simulated data using CNNs. The sole target under consideration was the lobster trap. The measured SAS sensor data used in the study consists of 650 training samples and 345 test samples, and the PoSSM-generated data parameterized to simulate the SAS sensor consists o 231 training samples and 252 test samples.

3.1.2 CNNs

For each of the two sensors, three CNNs of varying network capacity were trained using the training data. The binary classification task for each CNN was to discriminate measured targets from simulated targets. The ability of the trained CNNs to perform this task was evaluated using the test data. When a CNN is able to successfully discriminate measured data from simulated data, it is an indication that the fidelity of details in the simulated data do not match the measured data. Conversely, when a CNN *cannot* reliably tell the difference between measured data and simulated data, it is an indication that the latter is of very high fidelity. The CNNs were designed such that their capacities differed, so that performance could be assessed as a function of CNN complexity.

The three CNNs shared a common general architecture, but differed most notably in the fact that a pre-pooling layer was employed directly after the input, which effectively reduced the resolution and detail of the imagery. When larger pre-pooling factors were used, the CNN would have fewer fine details to leverage during the training process to help identify the more subtle deviations between the measured and simulated data. An example chip with three different pre-pooling factors applied is shown in Figure 3.

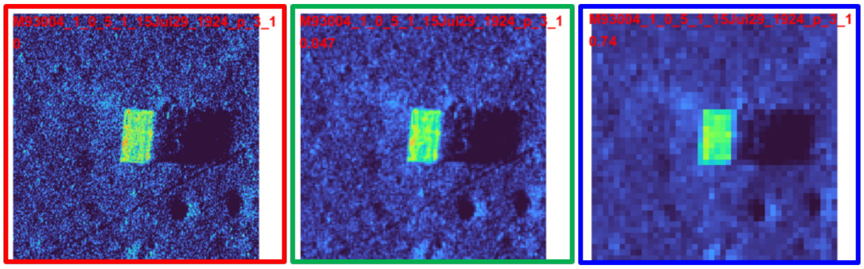

M93004_1_0_5_1_15Jul29_1924_p_3_1
0
M93004_1_0_5_1_15Jul29_1924_p_3_1
0.047
M93004_1_0_5_1_15Jul29_1924_p_3_1
0.74

Figure 3: Effective resolution induced by different CNN pre-pooling factors (from left to right, of 1, 4, and 10) on a SAS image of a lobster trap.

After the pre-pooling layer, the CNNs used alternating convolutional *blocks* – comprising one or more convolutional layers – and pooling layers. A high-level schematic of this architecture is shown in Figure 4. The input to the CNN is assumed to be a 300 pixel × 300 pixel SAS magnitude image chip, where each pixel spans 1.5 cm in each dimension. The outputs of the CNN's final layer are the probabilities of an image belonging to each class.

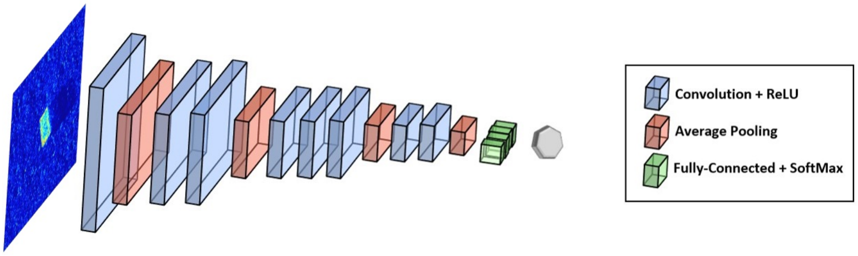


Figure 4: High-level illustration of the* PP1-C8 *CNN architecture. The input is a SAS magnitude image, and the output is a scalar prediction of class membership.

Each convolutional layer uses 4 square filters of stride 1 without padding and ends with a rectified linear unit (ReLU) activation function. A softmax activation is used at the output. Average pooling is used. Table 1 provides CNN design details, where brackets are used to convey convolutional *blocks*, in which there are multiple convolutional layers in between pooling layers. The convolutional block construct allows deeper networks, and thus greater complexity.

CNN training is performed in Python with the TensorFlow[8] software library. Training uses an RMSprop optimizer with a learning rate of $\eta = 0.001$, in conjunction with a binary-cross-entropy loss function. A batch size of $B = 64$ is used, with equal numbers selected from each class. Training was conducted for 10 epochs, where one epoch is defined to be a set of 1000 batches. Each batch is formed by randomly selecting samples from the full set of training data. Data augmentation that respects the geometry of the sonar data-collection procedure is employed during training; this means a random range translation $i_{tx} \in [-0.5 \text{ m}, 0.5 \text{ m}]$, along-track translation $i_{ty} \in [-0.5 \text{ m}, 0.5 \text{ m}]$, and along-track reflection $i_{ry} \in \{\pm 1\}$ is applied, "on-the-fly," to each SAS image chip selected for the batch.

Table 1: Architectures for the CNNs

CNN Label	CNN Depth	Pre-Pooling Factor	Filter Sizes (Pixels Per Side)	Pooling Factors	Number of Parameters
PP1-C8	8	1	$\begin{bmatrix}9\end{bmatrix}\begin{bmatrix}6\\3\end{bmatrix}\begin{bmatrix}5\\3\\3\end{bmatrix}\begin{bmatrix}4\\3\end{bmatrix}$	4 3 2 2	2169
PP4-C3	3	4	$\begin{bmatrix}8\end{bmatrix}\begin{bmatrix}6\end{bmatrix}\begin{bmatrix}5\end{bmatrix}$	4 2 2	1249
PP10-C2	2	10	$\begin{bmatrix}11\end{bmatrix}\begin{bmatrix}3\end{bmatrix}$	5 2	641

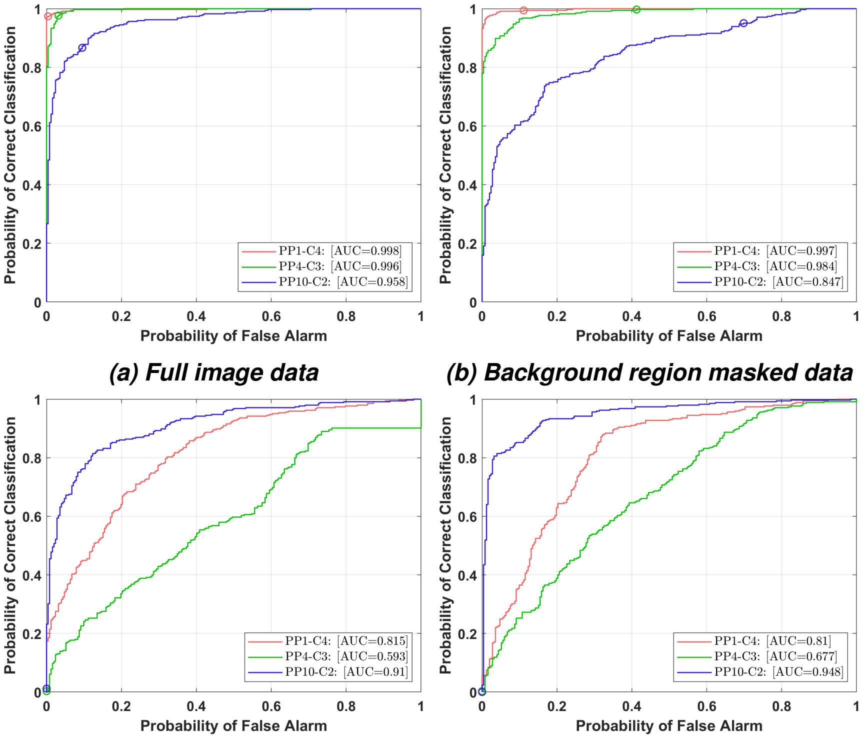
3.1.3 Results

The performance of the trained CNNs on the test data sets is shown in Figure 5; the operating point corresponding to a 0.5 decision threshold is marked with a circle on each curve. For purposes of ROC-curve presentation, the measured data was treated as the target class, and the simulated data was treated as the clutter class. Given that the classification task was to discriminate measured data from simulated data, worse ROC curves imply scenarios where the two classes of data appeared more similar to the CNN. The cases for which a CNN could reliably distinguish the two classes of data correspond to instances of lower model fidelity.

From the figure, it can be observed that classification performance degrades as the network capacity decreases and the pre-pooling factor increases. That is, as lower-resolution imagery is provided to the CNNs, there are fewer reliable differences between the measured and simulated data for the CNN to leverage for discrimination.

To better understand what features the CNNs were relying on to make classification decisions, an ablation study was undertaken with three different masks. One masked the background, one masked the center target (highlight) portion, and one masked both the target portion as well as the expected shadow region. The masks were of fixed size for all chips. The pixel values substituted in a given masked region were drawn from a normal distribution with mean set to the median pixel value of the original chip, and a standard deviation of 0.25 (the chips' pixel values are in $[-1, 1]$). These values were used rather than setting the pixel values to zero in order to minimize the information content in the masked regions.

The three masked versions of each test image were created and evaluated by the trained CNNs. The results of these ablation studies are shown in Figures 5 (b), (c), and (d). When the background region is masked, in Figure 5 (b), the CNNs' discrimination ability worsens, indicating that some differences in background between the measured and simulated data were being leveraged by the CNNs. But when the target portions of the imagery were masked, performance dropped more, indicating that the target region was the main source of information being used by the CNNs. These results help identify parts of the simulated data that exhibit a larger difference from the measured data, for which further modeling effort is required.



(a) Full image data

(b) Background region masked data

(c) Target region masked data

(d) Target and shadow region masked

Figure 5: Ability of the CNNs to discriminate measured data from simulated data, (a) with the full image was under consideration, (b) with the background regions were masked, (c) with the target (highlight) regions masked, and (d) with target and shadow regions maksed.

3.2 Augmenting ATR Training With Simulated Data

3.2.1 Data

The objective of this experiment was to attempt to discriminate targets from non-targets (*i.e.,* clutter). The target class was lobster traps, while all other objects were treated as clutter. A mixture of measured data and simulated data were used for training, and performance was evaluated on a separate set of measured data. The measured non-target data sets used in the study is from four U.S.-based collection sites, B, P, C, F, each contributing 33,561, 53,321, 170,658, and 28,775 samples, respectively. The total number of non-targets is 286,405. The measured target data used for training varied between [0,400], and the measured test data consisted of 345 targets.

For the experiments with measured data, the *non-target* data were divided into four collections by site. For a given case, three of the collections would be used for training, while the fourth would be used for testing. Each collection was treated as the test set for one case, allowing performance to be observed as a function of site background.

Because the vast majority (around 85%) of the *target* examples come from a single data set, namely Collection B, the target data were divided into a single fixed training pool and a single fixed test set, which did not vary across different cases. From the training pool, subsets of $N_m = \{0, 4, 20, 40, 100, 200, 300, 360, 400\}$

measured target training examples were created. The specific training examples selected for a subset were randomly chosen, with the caveat that all training examples selected for a smaller subset were also included in every larger subset.

The target training data for a given subset was supplemented with PoSSM-generated data parameterized to simulate the SAS sensor. Specifically, $N_s = N_t - N_m$ simulated target examples were added to the measured data subsets, with $N_t = 400$. That is, for each subset of data, the total number of target examples was fixed at 400, but the proportions of measured and simulated examples within each varied. The choice of setting $N_t = 400$ was based on the observation that a training set of 400 target examples appeared to be sufficient for performance to plateau.

Let x denote the fraction of targets in a given subset of training data that are *measured* data; thus $x = 0, 0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 1.0$ were the fractions considered. The $x = 0$ case corresponds to a training data set consisting of only simulated target examples; the $x = 1$ case corresponds to a training data set consisting of only measured target examples. The $x = 0.01$ case corresponds to a training data set consisting of 4 measured target examples and 396 simulated target examples.

The experimental set-up outlined above will reveal how performance varies as the proportions of measured and simulated target data in the training set are altered (for a fixed number of total target examples). However, it will not answer what performance would have been if *only* the measured target data was used. Therefore, the aforementioned experiments will also be repeated with the given amounts of measured target data, but with *no* simulated target examples. This arrangement addresses the question: for a fixed number of measured target examples, does adding simulated data help?

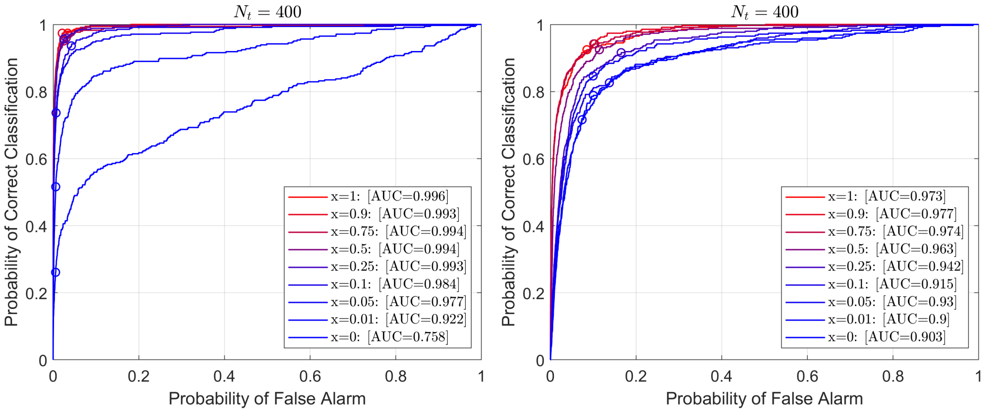
3.2.2 CNNs

The binary classification task for a given CNN was to discriminate targets from non-targets. The ability of the trained CNNs to perform this task was evaluated using the test data, which consisted only of measured data. The CNNs were trained using training data, which consisted of measured non-target data, but mixtures of measured and simulated target data. Four CNNs were considered, three of which were described in Table 1. An additional CNN, PP1-C22R, used a pre-pooling factor of 1 and contained residual layers, and therefore had considerable convolutional depth, with 3941 learnable parameters. The CNNs were designed such that their capacities differed, so that performance could be assessed as a function of CNN complexity. The training procedure mirrored that described in Sec. 3.1.2.

3.2.3 Results

The performance of the trained CNNs in discriminating targets from non-targets on the test data sets is shown in Figure 6; the operating point corresponding to a 0.5 decision threshold is marked with a circle on each curve. In the figures, the gap between the red curve $(x = 1)$ and the blue curve $(x = 0)$ corresponds to the drop in performance from using simulated data instead of measured data. This gap, then, can be viewed as a measure of model-fidelity error. In general, when higher-capacity CNNs are employed, the gap in performance is larger. That is, these CNNs are impacted more by model-fidelity limitations. Conversely, with lower-capacity CNNs, there is less variance in performance as the fraction of simulated data in the training set is varied. In all cases, as expected, the use of all measured data achieves better performance than a mixture of measured and simulated data.

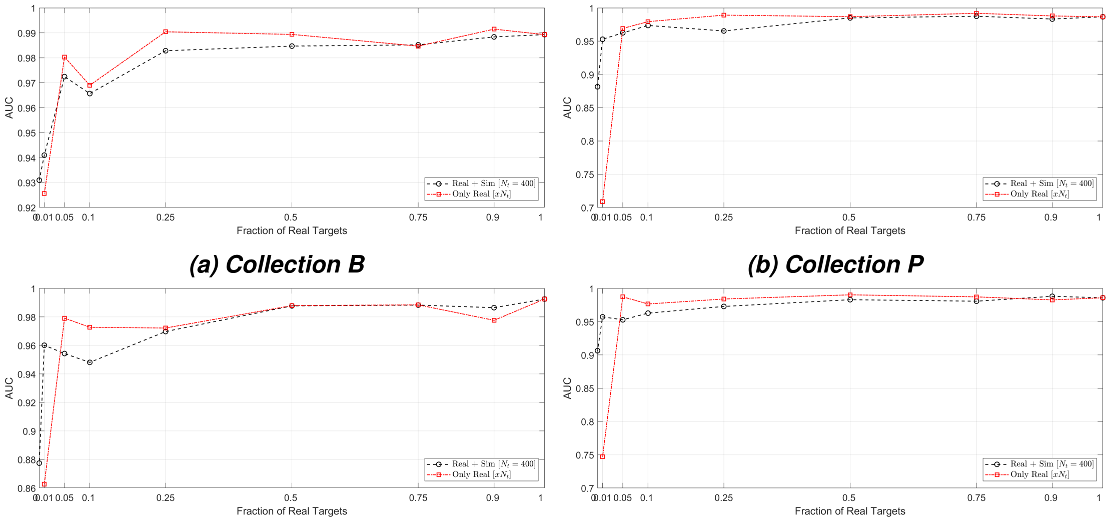
Next, classification performance for measured data is assessed in the form of area under the ROC curve



***(a) CNN* PP1-C22R** ***(b) CNN* PP10-C2**

Figure 6: For the measured data, ability of different CNNs to discriminate targets from Collection B non-targets, as x, the fraction of targets in the training set that is* measured*, varies. The ROC curves (a) exhibits a wide spread of performance, and a stronger dependence on x, whereas in (b) the dependence is weaker. The ROC curves from CNNs PP1-C8 and PP4-C3 also follow this trend.

(AUC) when (i) a mixture of measured (*i.e.,* “real”) and simulated data is used for training, and (ii) the same amount of measured data is used without any additional simulated data. These results are shown in Figure 7. The black curves correspond to the former case, while the red curve corresponds to the latter case. In these figures, at a given value on the x-axis, which determines the number of measured targets in the training set, a black point above the corresponding red point indicates that performance improved with the addition of simulated data.



(a) Collection B ***(b) Collection P***

(c) Collection C ***(d) Collection F***

Figure 7: For the measured data, ability of CNN* PP4-C3 *to discriminate targets from different collection non-targets, as the number of measured targets in the training set varies, with (black, dashed line) and without (red, solid line) supplemental simulated target examples.

In general, with a high-capacity CNN, adding simulated data hurts performance. That is, training with fewer target examples is preferable to training with additional target examples that are simulated. With lower-capacity CNNs, supplementing measured target examples with simulated target examples tends to improve performance only when the number of measured target examples is very low, like 0 or 4: what we consider data-starved scenarios. We hypothesize that if the model fidelity of the simulated data improved, training with simulated data would be beneficial for cases with more measured target examples.

4 CONCLUSIONS

There is an increasing need for modeling and simulation for machine learning in undersea applications. Simulated data can be used to supplement ATR training for robust performance in environments for which real survey data is not available. Lobster traps were used as the objects of interest for this analysis. In this work, various physically consistent factors of realism were incorporated during object modeling and signal processing in order to generate realistic simulated acoustic data suitable for synthetic aperture sonar (SAS) image reconstruction. The utility of that realistic simulated acoustic data and reconstructed imagery was then quantified through the lens of machine learning in two ways. First, convolutional neural networks (CNNs) were trained specifically to discriminate simulated data from field data. It was demonstrated that the misclassification rate of the classifier increases with the level of realism of the simulated data. Second, CNN-based ATRs were trained using a mixture of simulated data and field data, with varying ratios of each, and then tested on unseen measured data. Results showed that the utility of simulated data is highest in data-starved scenarios, when relatively little measured data is available for training.

REFERENCES

1. J. Stack, Automation for underwater mine recognition: current trends and future strategy, *Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XVI*, International Society for Optics and Photonics (2011).
2. S. F. Johnson, A. P. Lyons, ”Simulation of rippled-sand synthetic aperture sonar imagery.” *Institute of Acoustics Proceedings* 32.pt 4 (2010).
3. D. C. Brown, S. F. Johnson, D. R. Olson, ”A point-based scattering model for the incoherent component of the scattered field.” *The Journal of the Acoustical Society of America* 141.3: EL210-EL215 (2017).
4. S. F. Johnson, D. C. Brown, ”SAS simulations with procedural texture and the point-based sonar scattering model.” *OCEANS MTS/IEEE* Charleston. IEEE, (2018).
5. A. T. Abawi, ”Kirchhoff scattering from non-penetrable targets modeled as an assembly of triangular facets.” *The Journal of the Acoustical Society of America* 140.3: 1878-1886 (2016).
6. I. D. Gerg, D. C. Brown, S. G. Wagner, D. Cook, B. O'Donnell, T. Benson, T. C. Montgomery, ”GPU acceleration for synthetic aperture sonar image reconstruction.” *Global OCEANS*: Singapore–US Gulf Coast. IEEE, (2020).
7. J. D. Park, G. Geohle, B. Cowen, T. E. Blanford, D. C. Brown, ”Assessing the Utility of Multiple Representations for Object Classification.” *IEEE Transactions on Geoscience and Remote Sensing* (2023).
8. M. Abadi, et. al., ”TensorFlow: Large-scale machine learning on heterogeneous systems.” (2015).