Multi-Label Classification of Heterogeneous Underwater Soundscapes with Bayesian Deep Learning

Marko Orescanin [1], Brandon Beckler [2], Andrew Pfau [2], Sabrina Atchley [2], Nicholas Villemez [2], John Joseph [2], Christopher Miller [2], and Tetyana Margolina [2]

[1]Naval Postgraduate School
[2]Affiliation not available

October 30, 2023

Posted on 3 Mar 2020 — CC-BY 4.0 — https://doi.org/10.36227/techrxiv.19265016.v1 — e-Prints posted on TechRxiv are preliminary reports that are not peer reviewed. They should not b...

Multi-Label Classification of Heterogeneous Underwater Soundscapes with Bayesian Deep Learning

Brandon Beckler, Andrew Pfau, Marko Orescanin, *Member, IEEE,* Sabrina Atchley, Nicholas Villemez, John E. Joseph, Christopher W. Miller, and Tetyana Margolina

Abstract

Underwater soundscapes of coastal zones close to human settlements are heterogeneous in nature. Multiple ships and biological sources are often simultaneously present in the passive sonar vicinity. Classification of such heterogeneous underwater soundscapes is a challenging task for humans as well as machine learning systems. In this work a Bayesian Deep Learning approach is proposed that can accurately classify multiple ships simultaneously present in the vicinity of the sensor (multi-label classification) and provide uncertainty in the classification. This is achieved by assuming a Bayesian formulation of standard convolutional neural network architectures to not only assign multi-labels per inference but also to provide per inference uncertainty. By utilizing almost 3,500 hours of passive sonar data (spanning more than a year of sensor deployment) labeled through automated fusion with automatic identification system information, both multi-class and multi-label classification tasks of ship-generated noise are addressed. The best performing Bayesian architecture on the multi-label task achieves a weighted F^1 score of 0.84, where each prediction is accompanied by a measurement of uncertainty which is used to further enhance the understanding of model predictions.

Brandon Beckler, Marko Orescanin, and Sabrina Atchley are with the Department of Computer Sciences, Naval Postgraduate School, Monterey, CA, 93943 USA (e-mail: brandon.beckler@nps.edu; marko.orescanin@nps.edu).

Andrew Pfau is with the Computer Science Department, United States Naval Academy, Annapolis, MD, 21402 USA (e-mail: pfau@usna.edu)

Nicholas Villemez is with the Department of Information Sciences, Naval Postgraduate School, Monterey, CA, 93943 USA (e-mail: nicholas.villemez@nps.edu)

John E. Joseph, Christopher W. Miller and Tetyana Margolina are with the Department of Oceanography, Naval Postgraduate School, Monterey, CA, 93943 USA (e-mail: jejoseph@nps.edu; cwmiller@nps.edu; tmargoli@nps.edu).

Multi-Label Classification of Heterogeneous Underwater Soundscapes with Bayesian Deep Learning

I. Introduction

The classification of underwater soundscapes is of interest to several communities, including biologists and oceanographers who look to study fish and whale populations through recordings from the underwater environment [1]–[3]. Shipping noise can have adverse impacts on marine mammal populations. The measurement and modeling of shipping noise is important in predicting environmental impacts. Such research aids autonomous monitoring of fisheries and fishery enforcement by government and environmental groups [4]. Ships, submarines, and unmanned underwater vehicles can use passive sonar classification systems to aid in the identification and tracking of contacts, to help maintain safety of navigation, to aid in the real-time interdiction of illicit activities (such as smuggling and covert vessel transits), and to provide port security [5], [6]. The use of machine learning algorithms for the classification of underwater sounds is well established [7]. Most research in this area, however, focuses on identification of biological sounds [1], [3] with considerably less reported research on man-made or ship sounds.

This lack of research is partly due to the fact that the datasets used in ship classification tasks are often limited, either in size or in similarity to real-world conditions. Zak used sounds recorded from just five naval vessels to demonstrate the use of self-organizing maps and neural networks to classify ship sounds with greater than 70% accuracy [8]. Santos-Domínguez et al. report using only two hours of recordings [9], and Niu et al. use just three ships with 30 minutes of recording from each ship [10]. Berg et al. [5] and Neilsen et al. [11] both use synthetically generated samples for training due to a lack of real-world data.

An overall lack of data also affects the quality of results by reducing the diversity of conditions in which ship noise is recorded. A ship on the ocean creates acoustic signals from operating

machinery, propeller cavitation, and the motion of propeller shafts and reduction gears [12]. Vibrations of operating engines and pumps are transferred through the hull into the water, creating a distinctive pattern of sound that can be detected by a hydrophone. The size, speed, and aspect to the sensor all affect the type and strength of signals received [12], as do oceanographic conditions such as temperature, salinity and pressure (primarily a function of depth) [13]. These conditions change regularly depending on factors such as weather, time of day, and time of year.

Arveson and Vendittis provide an overview of the sound sources and source levels that are generated by a bulk cargo ship [14]. McKenna et al. examined recordings of multiple commercial ships which show that the sound from container ships predominately falls below 40 Hz and that all ships showed asymmetry in their signatures, with bow aspect radiated noise lower than stern aspect [13]. These studies illustrate some of the challenges of automatic classification of ships, including differences in emitted noise from the same ship due to changes in equipment use, variable water conditions that can change how emitted sound from the same ship is picked up by the receiver, and changes in ship aspect and/or range relative to the receiver.

So far, underwater soundscape classification tasks have been treated as acoustic event classification, in which a sample contains a single acoustic event to be labeled with one out of a number of possible classes (this is known as multi-class classification) [1], [2]. This approach, however, is an inaccurate representation of the heterogeneous underwater acoustic environment where multiple ship signals are often simultaneously present. Machine learning models trained on a multi-class classification task will provide a single label to the input data stream and will miss labeling any other ships present in the audio sample. The ability to demonstrate underwater soundscape classification on multiple, simultaneous ships using a single element hydrophone (measuring scalar pressure only) and provide an uncertainty measurement for those estimates has remained a challenge for the community at large.

This paper addresses that challenge by using a multi-label classification model, which has the ability to assign one or more labels to an individual sample. In contrast to the common approach of rare acoustic event classification, here the goal is to detect multiple target labels per inference (per sample) of the neural network classifier. Similar to the Google YouTube8M challenge [15], this is achieved by developing and evaluating a multi-label convolutional neural

network (CNN) architecture. To address the lack of uncertainty measurement in the classification estimates, a Bayesian Deep Learning (BDL) approach is adopted. BDL blends deep learning with Bayesian theory to enable models that provide uncertainty measurements and are more robust to overfitting relative to the deterministic (classical) neural network architectures [16]. Uncertainty measurements allow for a deeper understanding of the model's predictions [17].

In this work, BDL model architectures are developed to not only establish the link between the ship acoustic signature and the classification ontology adopted, but also to estimate the uncertainty in the classification. Both deterministic and Bayesian configurations of deep Residual Networks (ResNet) model [18], [19] architectures and a custom CNN architecture are adopted and benchmarked on the task. The advantage of the Bayesian architecture over its deterministic counterpart is outlined and uncertainty of inference of ship classification is presented as a unique improvement of Bayesian architectures for underwater soundscape classification applications over deterministic counterparts. Moreover, the large size of our dataset, with more than 4,000 unique ships and over 3,400 hours of labeled audio data, enabled us to study the impact of seasonal variations of sound speed profiles (SSPs) on the bias and quality of classifications of developed deep learning models. Additionally, we study the quality of the measured uncertainty through a use-case study and correlate the uncertainty of classification to distance from the sensor and the bow-stern orientation.

II. Dataset

The dataset used for model training and evaluation was recorded at Thirty Mile Bank off the coast of southern California from December 2012 to November 2013, totaling more than 6,800 hours of recordings with 4,470 unique ships recorded. The sensor, a High-frequency Acoustic Recording Package (HARP), was deployed in 734 meters of water with the sensor 51 m above the sea floor and an original sample rate of 200 kHz [20]. Recordings were downsampled to a 4 kHz sample rate for labeling and model training [12].

In parallel to the HARP deployment, we used automatic identification system (AIS) data to develop datasets for both multi-class and multi-label tasks [12]. Given that there is no formal ontology of ship sounds, in order to utilize the AIS stream this research expands upon the ship

ontology described by Santos-Domínguez et al. [9], in which ships are divided into four classes based upon size and one class is given to samples without ship sounds (see Table I). Having a no-ship class enables the development of a flat-classifier instead of using a multi-level classifier, which would utilize a detector of ship presence followed by the classification algorithm. For the task of multi-class classification, 30 second audio samples were only assigned one label indicating which class of ship was present based on AIS messages [12]. In contrast, for the multi-labeled dataset, 30 second audio samples with more than one ship present were labeled with the class labels of all ships present at that time. Ship class was determined by matching audio data segments based on timestamps with AIS messages. Specifically, broadcast Maritime Mobile Service Identity (MMSI) numbers (or International Maritime Organization (IMO) numbers where MMSI number could not be found) allowed for finding precise ship details in an online ship database [21]. For both datasets, a ship was deemed present if within 20 km (10.7 NM) of the sensor; time periods where all ships were outside 30 km from the sensor were labeled as the no-ship class. For the multi-labeled dataset, samples with more than one ship present were labeled with the class labels of all ships present at that time within 20 km [12]. Only 33% (136,044 of 415,951 samples) of the dataset contained samples with more than one ship present, which is a common data imbalance in multi-label classification for audio [15].

Class	Ship Designators
A	Fishing Vessel, Tug, Towing Vessel
B	Pleasure Craft, Sailboat, Pilot
C	Passenger ship, Cruise Ship
D	Tanker, Container Ship, Military Ship, Bulk Carrier
E	No ship present, background noise

TABLE I
SHIP CLASSES

To produce intermediate signal representations used for training we use well-established, low-level acoustic signal representations in the form of mel-log spectrograms. Mel-log spectrograms are dominant features in deep learning [22] and are related to the linear-frequency spectrogram, i.e., a Short Time Fourier Transform (STFT) magnitude. They are obtained by applying a mel-

filterbank over STFT magnitude which effectively summarizes frequency content with fewer dimensions [23]. The mel-filterbank emphasizes details in lower frequencies, which were proven to be important in underwater soundscape classification, and deemphasizes higher frequency content which generally does not need to be modeled with high fidelity [13].

Specifically, in this work mel-log spectrograms were computed through a STFT of 30 second labeled samples with a 250 ms frame size, 75 ms frame hop, and a Hann window function [12]. We transform STFT magnitude to the mel-scale using a 128 band mel-filterbank followed by log compression of the signal [23]. Labeled samples for both tasks were further split into training/validation/test data with an 80/10/10 ratio, respectively.

In Fig.1 we visualize mel-log spectrograms for several examples, including having only a single target present and having two targets present at the same time. The color bar represents a dB down-scale. Relationship between mel scale and frequency [23] is given with $\text{mel}(f) = 2595 * \log_{10}(1 + \frac{f}{700})$. The sheer drop-off in power at 2000 Hz (1521 mel, top of each mel-log spectrogram) is related to anti-aliasing filtering and downsampling of the signal. Through visual inspection one can observe differences in input features and recognize the challenge of having two classes present in a single mel-log spectra given the diversity of target signatures and underlying variability of propagation paths from the target to the sensor.

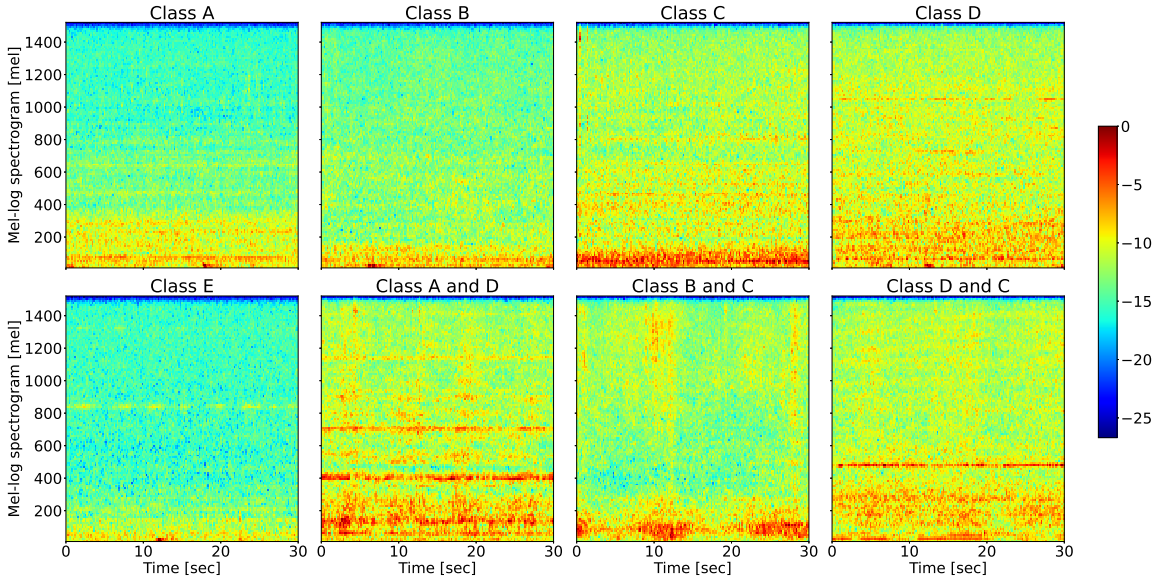


Fig. 1. Examples of mel-log input features from the dataset. Color bar represents db-down scale

The US Navy's Generalized Digital Environmental Model (GDEM) product database provides global, gridded, steady-state ocean temperature and salinity profiles [24]. Monthly temperature and salinity profiles were extracted at the nearest GDEM point (32.5N 117.75W) to the HARP's location (32.666N 117.707W). These were used to derive SSPs [25] over the 12-month deployment period, see Fig.2.

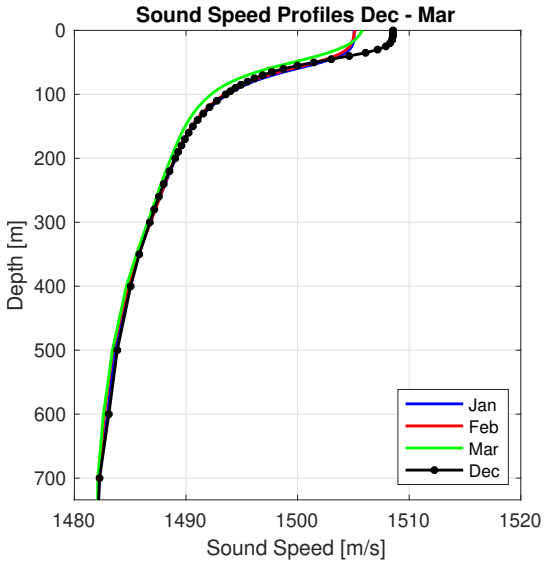
In order to evaluate the impact of environmental parameters on the performance of the trained models, two studies are conducted involving different data splits for the multi-class classification task, from December 2012 to March 2013 and from December 2012 to November 2013. From Fig.2a, from December to March low dispersion of the SSPs is observed, while for the overall time segment in Fig. 2b dispersion between the SSPs is significantly increased.

III. Methodology

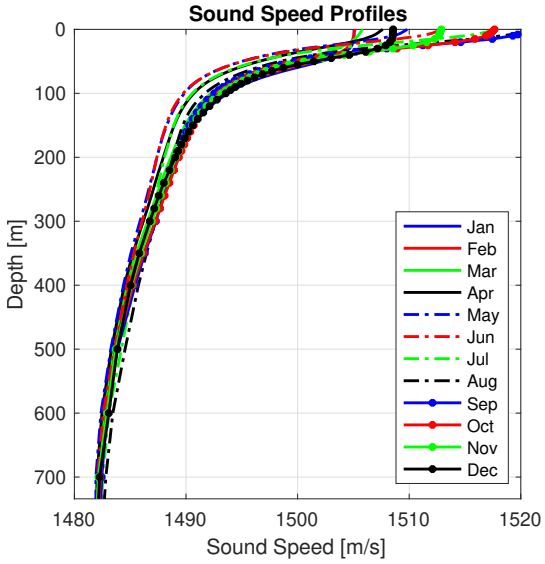
A. Bayesian Deep Learning

BDL combines the ability of Bayesian probabilistic models to provide uncertainty in predictions with the ability of neural networks to recognize patterns and relationships [17], [26]. Specifically, model uncertainty is measured by placing a prior probability distribution over the model's weights in order to construct a Bayesian CNN (BCNN) [27]. Given a supervised learning setting and a training dataset, $\mathcal{D} = \{\boldsymbol{x_n}, y_n\}_{n=1}^{N}$, where N represents the dataset size, $\boldsymbol{x_n}$ represents an input feature vector (where $\boldsymbol{x_n} \in \mathcal{R}^m = [x_{1,n}, x_{2,n}, \ldots, x_{m,n}]$) and y_n represents the corresponding label (where $y_n \in \{1, 2, \ldots, C\}$; C being the number of classes), a neural network model's posterior goal is to estimate $\hat{y}_n = f(\boldsymbol{x_n})$. A neural network model with L layers is parametrized by the set of weights $\mathbf{w} = \{\mathrm{W}_i\}_{i=1}^{L}$. The Bayesian approach assumes a prior distribution over neural network parameters $p(\mathbf{w})$, with the goal of quantifying posterior uncertainty over the network parameters $p(\mathbf{w} \mid \mathcal{D})$ given a dataset $\mathcal{D}$. This prior distribution represents our assumption as to which functions of neural network parameters were likely to generate our data. In inference, one can calculate probability of the model prediction $\hat{y}$ on a test data input $\boldsymbol{x}^*$ by integrating over all possible values in $\mathbf{w}$:

$$p\left(\hat{y}|\boldsymbol{x}^*, \mathcal{D}\right) = \mathbb{E}_{p(\mathbf{w}|\mathcal{D})}\left[p\left(\hat{y}|\boldsymbol{x}^*, \mathbf{w}\right)\right] = \int_{\mathbf{w}} p\left(\hat{y}|\boldsymbol{x}^*, \mathbf{w}\right) p\left(\mathbf{w}|\mathcal{D}\right) d\mathbf{w} \tag{1}$$



(a) December to March



(b) December to November

Fig. 2. Monthly sound speed profiles at the nearest GDEM point (32.5N 117.75W) to the HARP's location (32.666N 117.707W). In 2a sound speed profile is illustrated for Dec-Mar time frame and in 2b sound speed profiles are illustrated for Dec-Nov months. Significant dispersion can be observed in 2b relative to 2a.

In practice, because inference defined in Eq. 1 is intractable due to calculation of the probability distribution $p(\mathbf{w} \mid \mathcal{D})$, an approximate inference is used.

In this work, we evaluate variational inference (VI) approaches [26], [28], [29] that approximate the posterior distribution $p(\mathbf{w} \mid \mathcal{D}) \propto p(\mathbf{w})p(\mathcal{D} \mid \mathbf{w})$ by fitting an approximation $q_\theta(\mathbf{w}) \approx p(\mathbf{w} \mid \mathcal{D})$, where θ are the parameters of the probability distribution over weights. This

approximate distribution needs to be as close as possible to the posterior distribution. A common approach in information theory is to measure proximity (or similarity) between two distributions via Kullback-Leibler (KL) divergence [30]. Therefore, we minimize the KL divergence between two distributions:

$$\mathrm{KL}\big(q_\theta(\mathbf{w}) \,||\, p(\mathbf{w} \mid \mathcal{D})\big). \tag{2}$$

Minimizing the KL divergence in Eq. 2 is equivalent to minimizing the negative evidence lower bound function (ELBO) [16], [28], [31] relative to θ:

$$\begin{aligned}\mathcal{L}(\theta) = -\mathbb{E}_{q_\theta(\mathbf{w})}\big[\log p\big(\mathcal{D} \mid \mathbf{w}\big)\big] \\ + \mathrm{KL}\big(q_\theta(\mathbf{w}) \,||\, p(\mathbf{w})\big)\end{aligned} \tag{3}$$

where the first term represents the expected likelihood, which "describes how the variational distributions of the neural parameters explain the observed data," [32] and the second term is KL divergence measuring proximity between the posterior and prior densities. This cost function definied by Eq. 3 is minimized in a mini-batch stochastic gradient descent fashion during neural network training to find the optimal value of θ which defines the parameters of the distribution over weights.

In order to optimize the lower bound with respect to variational parameters θ and model parameters $\mathbf{w}$, the expectation term in Eq. 3 must be differentiated with respect to both of these variables. The problem of computing the gradient of an expectation of a function with respect to the parameters of the distribution is addressed by Monte Carlo gradient estimation [33]. However, the standard Monte Carlo gradient estimator of the variational lower bound with respect to the variational parameters, θ, exhibits a level of variance that is too high to be practical for deep learning purposes [28], [34]. Proposed solutions are based on the reparameterization of $q_\theta(\mathbf{w})$ such that the samples generated from the reparameterized approximate posterior yield a lower variance [28], [30], [31]. Some of the well-established solutions that are part of major toolboxes such as Tensorflow Probability (TFP) [35] include techniques such as the local reparametrization

trick (LRT) [30] and flipout estimators [31]. While both estimators present an improvement over standard variational inference, the flipout estimator, given several assumptions and a trade-off in computational complexity, can yield decorrelated stochastic gradient estimates on a mini-batch that exhibit less variance than the estimated mini-batch gradients computed via an LRT approach [31]. Intuitively, from an optimization perspective, decorrelation leads to better conditioning of the Hessian for updating the weights by whitening the external and internal network-layer inputs. In our evaluations we used a default TFP implementation of 2D Convolutional flipout layers which assumes a Gaussian distribution with variational parameters $\theta = (\mu, \sigma)$, variational distribution $q_\theta(\mathbf{w}) = \mathcal{N}(\mu, \sigma^2)$ and a prior with $p(\mathbf{w}) = \mathcal{N}(0, 1)$.

Given that a flipout estimator effectively doubles the number of parameters of the deterministic neural network layer, we evaluated a simpler method that does not explicitly model distributions over weights called Monte Carlo (MC) Dropout [16], [27], [36], [37]. MC Dropout is based on a standard neural network regularization technique called dropout which was introduced by Srivastava et al. [38]. Dropout is applied as a layer in neural networks where it zeros out a random subset of the weights in the preceding layer during training only. It was shown [16], [39] that the set of mean weight matrices (L layered neural network) and dropout probabilities (variational parameters) for a dropout distribution satisfies $\theta = \{\mathbf{M}_l, p_l\}_{l=1}^{L}$, such that $q_\theta(\mathbf{w}) = \prod_l q_{\mathbf{M}_l}(\mathbf{w}_l)$ and $q_{\mathbf{M}_l}(\mathbf{w}_l) = \mathbf{M}_l \cdot [\text{Bernoulli}(1 - p_l)^{K_l}]$ for a single random weight matrix $\mathbf{w}_l$ of dimensions K_{l+1} by K_l. Further, the KL term of Eq. 3 can be approximated as shown in Eqs. 4 and 5:

$$\text{KL}\big(q_\theta(\mathbf{w}) \,||\, p(\mathbf{w})\big) = \sum_{l=1}^{L} KL\big(q_{\mathbf{M}_l}(\mathbf{w}_l) \,||\, p(\mathbf{w}_l)\big) \tag{4}$$

$$\text{KL}\big(q_{\mathbf{M}}(\mathbf{w}) \,||\, p(\mathbf{w})\big) \propto \frac{l^2(1-p)}{2} \|\mathbf{M}\|^2 - B\mathcal{H}(p) \tag{5}$$

where $\mathcal{H}(p)$ the entropy of a Bernoulli random variable with probability p [39] and B is the constant term related to the number of mini-batches in optimization. The entropy term is constant with respect to model weights and, given it does not affect the optimization, can be omitted when the dropout probability is not optimized. To calculate KL divergence, one only needs to evaluate the probability of the dropout, p, and a second norm on the weights $\|\mathbf{M}\|^2$.

It was shown by Gal and Gharmani [16] that any neural network with standard dropout and L2 regularization is equivalent to variational inference with the assumption that the dropout is active in inference. Explicitly optimizing the dropout probability in the entropy term leads to a different technique named Concrete Dropout [39] which we have not evaluated in this work.

We wanted to point out that the log-likelihood term of the ELBO loss function, see Eq.3, can be either calculated in an explicit manner or implicitly takes the form of categorical-cross entropy for multi-class classification with a softmax activation function on top of the flipout and MC Dropout architectures [40]. Similarly, for multi-label classification, the log-likelihood term takes the form of the binary-cross entropy with sigmoid activation. For readers interested in implementation details, Filos et al. [36] hosts a code repository that includes details of both flipout and MC Droput methods and we followed their implementation approach. Additionally, Google LLC hosts a code repository, see Nado et al. [41], with current state-of-the-art benchmarks in Bayesian Deep Learning, including methods presented in this manuscript.

For a trained neural network, with weights $\hat{\mathbf{w}}_t$, prediction uncertainty is induced by the uncertainty in weights and can be calculated by marginalizing over the approximate posterior distribution $q_\theta(\mathbf{w})$ using Monte Carlo integration [16] with T samples to calculate mean predictive probability from Eq. 1:

$$\begin{aligned} p(\hat{y} = c \mid \boldsymbol{x}^*, \mathcal{D}) &\approx \int p(\hat{y} = c \mid \boldsymbol{x}^*, \mathbf{w}) q_\theta(\mathbf{w}) d\mathbf{w} \\ &\approx \frac{1}{T} \sum_{t=1}^{T} p(\hat{y} = c \mid \boldsymbol{x}^*, \hat{\mathbf{w}}_t) \\ &\approx \frac{1}{T} \sum_{t=1}^{T} \hat{p}_{c_t} = \bar{p}_c \end{aligned} \tag{6}$$

where $\hat{\mathbf{w}}_t \sim q_\theta(\mathbf{w})$ and c represents the true class (e.g., "Class A" or "Class D"). Further, final classification is assigned based on Eq. 6 by assigning the class based on the highest mean predictive probability. It is important to clarify that, in inference, prediction is repeated T times for each input to the trained neural network for both flipout and MC Dropout. Intuitively, with every prediction for the given input a different set of weights is sampled from the model and

an ensemble of predictions is produced, with final prediction being an ensemble average. For further clarification, in contrast to dropout for regularization during training, in the MC Dropout approach, dropout is active with probability p during inference as well.

Input mel-log spectrograms were classified from Eq. 6 by assigning the class based on the highest mean probability, argmax $\bar{p}_c$. The uncertainty of BCNN classifiers is quantified by predictive entropy and total variance [16], [36]. Predictive entropy is a well-established uncertainty metric that measures the average amount of information contained in the predictive distribution and is given by:

$$H_p(\hat{y} \mid \boldsymbol{x}^*) = -\sum_c \bar{p}_c \log \bar{p}_c \tag{7}$$

H_p can be normalized to fall between zero and one by dividing by $\log 2^C$ (which comes out to C, the number of classes) [42] as shown in Eq. 8.

$$H_p^*(\hat{y} \mid \boldsymbol{x}^*) = -\sum_c \bar{p}_c \frac{\log \bar{p}_c}{\log 2^C} \tag{8}$$

B. Architecture Choices and Tasks

The popularity of CNNs is due to the state-of-the-art performance that these model achieve in large scale image recognition tasks [43]. A desire to train deeper neural networks, potentially improving performance further, led to the development of residual connections introduced by He et al. [18], who demonstrated the ability to train deeper convolutional models than previously possible. We utilize these ResNet architectures in parallel to the custom CNN architecture to train and develop deterministic and BDL models for classification.

In this work, we focus on two classification tasks, multi-class classification and multi-label classification. Multi-class classification assumes a multinoulli probability distribution since one wants to represent distribution over c classes. This is typically achieved by having c neurons in the last layer of the neural network and applying a softmax activation function, $\hat{p}_{c_t} = softmax(\hat{f}_t(x^*))$, see Eq.6.

Multi-label classification is typically formulated as multiple binary classification problems when using a negative log likelihood loss (cross-entropy loss) [44]. A similar approach was followed by Hershey et al. [15] for audio classification using c sigmoid activation functions (σ) one over each of the c output neurons:

$$\hat{p}_{c_t} = \sigma\left(\hat{f}_t(\boldsymbol{x}^*)\right) \tag{9}$$

where $\hat{f}_t(\boldsymbol{x}^*) \in \mathcal{R}^c$. This is a common approach in image multi-label classification [45], [46]. The choice of task drives the choice of activation function on the output of the neural network model, however, the number of output neurons is constant across the architectures.

Multiple labels can be predicted when the individual probabilities on the output neurons are greater than the probability threshold, in this work set at 0.5 [40], [44]. The threshold was not tuned to maximize any specific metric, such as F^1 score or to reduce the false-alarm rate. Since our ontology includes all ships and "no ship" as classes, in the case where none of the classes meet the threshold, we select the predicted class in the same manner as the multi-class classifier described above. In the case where both the "no ship" class, Class E, and at least one other class both meet the threshold, if the probability for Class E is greater than those for any other class, the model predicts Class E and nothing else. If at least one of the ship classes has a greater than or equal probability than that for Class E, the model predicts every ship class that meets the threshold and not Class E.

For ResNet [18] architectures, we tested standard ResNet32V1, ResNet20V1 and ResNet8V1 model configurations as a deterministic baseline that we adapted to Bayesian configurations following suggestions in Tran et al. and Dillon et al. [35], [47] Typically, CNNs that focus on image classification use square kernels of size 3x3 or 5x5; where larger kernels are considered inefficient due to computational requirements [40]. Assumptions about image orientation invariance do not transfer to spectrograms derived from time-series audio data [12], and multiple studies have explored the use of rectangular kernels in audio classification. Several studies use rectangular kernels in music classification [48]. Mars et al. use rectangular filters of various sizes to vary the convolution of time and frequency domains [49]. In order to adopt these ideas, rather

than adjusting a ResNet architecture, a custom CNN architecture is proposed as shown in Fig 3.

Through a hyperparameter search of kernel size ratios (1:1, 2:1, 3:1, and 4:1), using a kernel size of 5x5 as a baseline, it was found that a 2:1 ratio is optimal and a 4:1 ratio performed the worst [12] based on the model accuracy as the metric. Kernel ratios of 2:1 and 3:1 performed almost identically (with accuracy of 0.89) where a 4:1 ratio had a drop in performance (accuracy of 0.87) with a 1:1 kernel being in the middle (accuracy of 0.88). We chose 2:1 over 3:1 as optimal given that it introduces a lower number of parameters to the network architecture. Hence, the final proposed kernels of 10x5 were used to apply the convolution operation on time vs frequency mel-log spectrograms. Given our non-exhaustive ablation study for the kernel ratios, we did not change ResNet architectures and left them with the original isotropic kernel configurations of 3x3 as a fair and unbiased comparison throughout the manuscript. These kernel sizes are fixed throughout every layer of the network. A batch normalization layer is used to normalize input mel-log spectrograms. Based on work by Ozyildirim and Kartal [50], an increasing number of filters is used throughout the network. The initial layers contain 16 filters with 16 added in each additional set. After each block of two convolutional layers, the input size to the next block is cut in half by a max pooling layer with a stride of 2 by 2. L2-regularization on CNN layers is used to prevent overfitting [12] and to satisfy MC Dropout, see Eq. 5.

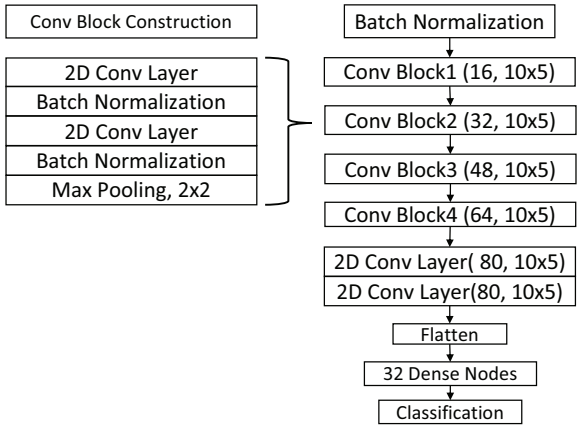


Fig. 3. Proposed Model Architecture: Each block is described by (number of filters, filter shape) [12].

C. Training Protocols

All of the model architectures evaluated were developed using the same training strategy for the fairness of benchmarking. Adam optimization was used with a starting learning rate of 0.001 and default values for beta parameters (0.9, 0.999). We employed learning rate annealing [51] via monitoring validation accuracy such that the learning rate was reduced by a factor of 10 if the validation accuracy was not increasing for 50 consecutive epochs. To regularize for overfitting, an early stopping strategy was utilized [40]. Overall, training was terminated at 500 epochs.

An MC Dropout model was used with a dropout probability of 0.3. Using a non-exhaustive grid parameter search, the best observed L2 regularization on convolutional layers was determined to be 0.001, and this value was used throughout all of the deterministic and MC Dropout layers. This is similar to Filos et al. [36]. For the flipout and initialization of the posterior, we followed TFP [35] recommendations for Bayesian ResNet architectures and have initialized convolutional kernel posteriors with μ=-9.0 and σ = 0.1. NVIDIA RTX 8000 48GB GPU graphics cards were used for distributed model training and model inference.

IV. Results

A. Metrics

In traditional binary classification tasks, standard metrics such as accuracy (Acc), precision ($Prec$), recall (Rec), F^1 score and area under the Receiver Operating Characteristics (ROC) curve are used to evaluate performance [40]. These metrics can also be extended to multi-class classification tasks in a fairly straightforward manner, since there is still only one label per sample. The performance is calculated per class and then averaged across all classes. This technique is known as macro-averaging. Averages can also be weighted by the number of instances of each label in the dataset, especially useful in the case of imbalanced datasets [52]. The ability of a single test instance to be associated with multiple labels simultaneously, however, makes multi-label performance evaluation much more complex than in the traditional single-label learning environment [53]. With multi-label models, micro-averaging is possible using the total number of true and false positives (TP and FP, respectively) and true and false negatives (TN and FN, respectively) to calculate the average globally. It is also important to note that any

of the multi-label metrics discussed below can be used to describe multi-class performance by treating the multi-class dataset as a multi-label one for which there happen to be no multi-label instances (micro-averaging for all multi-class metrics is equivalent to calculating accuracy).

There are two general categories of multi-label metrics: label-based metrics and instance-based (also called example-based or sample-based) metrics [53], [54]. Label-based metrics evaluate the machine learning model separately for each class label and then return either the micro- or macro-averaged value across all class labels, Eqs. 10 and 11. Given a testing dataset, $\mathcal{D}^* = \{(\boldsymbol{x}_i^*, Y_i)\}_{i=1}^{M}$, where M represents the test dataset size, $\boldsymbol{x_i}$ is the i -th feature vector (i.e., the i -th test sample) and Y_i is the set of true labels associated with the i -th test sample:

$$
\begin{aligned}
TP_j &= \{|\boldsymbol{x}_i^* \mid y_j \in Y_i \wedge y_j \in f(\boldsymbol{x}_i^*), 1 \leq i \leq M|\} \\
FP_j &= \{|\boldsymbol{x}_i^* \mid y_j \notin Y_i \wedge y_j \in f(\boldsymbol{x}_i^*), 1 \leq i \leq M|\} \\
TN_j &= \{|\boldsymbol{x}_i^* \mid y_j \notin Y_i \wedge y_j \notin f(\boldsymbol{x}_i^*), 1 \leq i \leq M|\} \\
FN_j &= \{|\boldsymbol{x}_i^* \mid y_j \in Y_i \wedge y_j \notin f(\boldsymbol{x}_i^*), 1 \leq i \leq M|\}
\end{aligned}
\tag{10}
$$

Eq. 10 describes how to calculate the value of TP, FP, TN and FN with respect to the j -th class label, y_j, where $f(\boldsymbol{x}_i^*)$ (or in case of the BDL $\hat{f}(\boldsymbol{x}_i^*)$) is the set of predicted labels output by the model f, or in case of BDL $\hat{f}$, on $\boldsymbol{x}_i^*$. Eq. 11 then describes how to use these values to compute traditional performance metrics using either macro- or micro-averaging, where $B \in \{Acc, Prec, Rec, F^1\}$ and C is the number of classes. Similarly, a macro-averaged area under the ROC curve (AUC) can be calculated by first computing the AUC for every class in a "one-vs-rest" manner and then averaging over C [53].

$$B_{micro} = B\left(\sum_{j=1}^{C} TP_j, \sum_{j=1}^{C} FP_j, \sum_{j=1}^{C} TN_j, \sum_{j=1}^{C} FN_j\right)$$

$$B_{macro} = \frac{1}{C}\sum_{j=1}^{C} B(TP_j, FP_j, TN_j, FN_j) \tag{11}$$

Since each instance for which the model makes predictions can be associated with more than one label, instance-based performance metrics can be aggregated by evaluating each test example individually and then averaging across the whole test set (in the multi-class case, the label-based and instance-based calculations are the same). For multi-label accuracy, we use subset accuracy as defined in Eq. 12, where $[\![q]\!]$ returns 1 if predicate q is true and 0 otherwise. This equates to the proportion of samples where the set of predicted labels for each sample, $f(\boldsymbol{x}_i^*)$ exactly matches the set of true labels, Y_i for the sample. This measure is intuitively the counterpart to traditional accuracy (the proportion of samples a model got "correct") and is the strictest measure of multi-label accuracy [53].

$$Acc_{subset} = \frac{1}{M}\sum_{i=1}^{M} [\![f(\boldsymbol{x}_i^*) = Y_i]\!] \tag{12}$$

The instanced-based methods of calculating other traditional performance metrics are listed in Eq. 13. Precision and recall are calculated for each instance by taking the size of the intersection of the set of true labels, Y_i, and the set of predicted labels $f(\boldsymbol{x}_i^*)$, or $\hat{f}(\boldsymbol{x}_i^*)$, divided by the size of the set of predicted labels or the size of the set of true labels, respectively. F^1 is then calculated in the usual way using the instance-based precision and recall.

$$Prec_{inst} = \frac{1}{M}\sum_{i=1}^{M}\frac{|Y_i \bigcap f(\boldsymbol{x}_i^*)|}{|f(\boldsymbol{x}_i^*)|}$$

$$Rec_{inst}(m) = \frac{1}{M}\sum_{i=1}^{M}\frac{|Y_i \bigcap f(\boldsymbol{x}_i^*)|}{|Y_i|} \tag{13}$$

$$F_{inst}^1 = \frac{2 \cdot Prec_{inst} \cdot Rec_{inst}}{Prec_{inst} + Rec_{inst}}$$

The final metric we discuss here is Hamming loss (HL), which evaluates the fraction of labels which are incorrectly predicted, that is, a relevant label is not predicted or an irrelevant label is predicted [53]. For each test instance, HL (Eq. 14) is the size of the symmetric difference, Δ (equivalent to XOR in Boolean logic), between the set of predicted labels, $f(\boldsymbol{x}_i^*)$, and the set of true labels, Y_i, divided by the number of classes, C [55]. The individual instance HLs are then averaged across all instances, and lower values indicate better performance. In the multi-class case, HL is equivalent to $1 - Acc$.

$$HL = \frac{1}{M}\sum_{i=1}^{M}\frac{1}{C}\,|f(\boldsymbol{x}_i^*)\Delta Y_i| \tag{14}$$

In this paper, in order to give a broad picture of the comparative performance of the several models tested, for both multi-class and multi-label models we report the macro-averaged and weighted-averaged label-based precision, recall, and F^1 score as calculated in Eq. 11, as well as the macro-averaged AUC and the HL (see Eq. 14). For multi-label performance, we also report label-based micro-averaged and instance-based precision, recall and F^1 score as shown in Eqs. 11 and 13. Accuracy is reported as discussed in the examination of Eqs. 11 and 12 above.

B. Performance Overview and Comparison

Our experiments examined the performance of traditional deterministic (non-Bayesian), MC Dropout, and flipout versions of both our custom CNN and ResNet models. For the ResNet models, the ResNet32V1 versions consistently performed better than the other ResNet configurations

tested. For example, weighted average F^1 scores for the multi-label MC Dropout versions of the model were 0.78, 0.76, and 0.71 for ResNet32V1, ResNet20V1 and ResNet8V1, respectively. Because this general pattern held across all versions, only the ResNet32V1 results are reported here. We trained models of each of the versions on the full HARP dataset for multi-class and multi-label classification, respectively. The results for multi-class classification are reported in Table II while the results for multi-label classification are in Table III. In both tasks, the models based on the custom CNN outperformed their ResNet model counterparts in every metric. In most cases, the best performing version of the custom CNN was the MC Dropout BCNN, generally reflective of the performance enhancements seen in ensemble learning (MC Dropout can be viewed as an ensemble classifier where each inference is a different model) [17]. The exception to this was in multi-label classification, where the deterministic and MC Dropout versions performed almost identically, varying at most by 2% in any one metric. We speculate, based on uncertainty measurements discussed below, that the greater predictive uncertainties involved in multi-label classification offset the performance gains often seen with ensemble learning by introducing enough variation that the MC Dropout model was unable to make more accurate predictions than the deterministic model.

	Custom			ResNet32v1		
	Det	Drop	Flip	Det	Drop	Flip
HL	0.193	**0.174**	0.204	0.284	0.248	0.279
AUC	0.967	**0.976**	0.964	0.910	0.938	0.916
Acc	0.807	**0.826**	0.796	0.716	0.752	0.721
$Prec_{macro}$	0.75	**0.78**	0.74	0.66	0.68	0.66
Rec_{macro}	0.72	**0.74**	0.70	0.58	0.65	0.61
F^1_{macro}	0.73	**0.76**	0.72	0.60	0.66	0.63
$Prec_{weight}$	0.80	**0.82**	0.79	0.71	0.75	0.72
Rec_{weight}	0.81	**0.83**	0.80	0.72	0.75	0.72
F^1_{weight}	0.80	**0.82**	0.79	0.71	0.75	0.71

TABLE II
MULTI-CLASS PERFORMANCE METRICS

To further examine the performance of multi-label classification, we present per class analysis in Table IV for both custom CNN and ResNet32v1 model architectures in Bayesian config-

	Custom			ResNet32v1		
	Det	Drop	Flip	Det	Drop	Flip
HL	**0.068**	0.071	0.089	0.117	0.096	0.120
AUC	**0.860**	0.848	0.814	0.770	0.797	0.763
Acc_{subset}	**0.743**	0.737	0.695	0.627	0.676	0.616
$Prec_{macro}$	**0.94**	**0.94**	0.88	0.76	0.85	0.75
Rec_{macro}	**0.74**	0.72	0.67	0.61	0.64	0.60
F^1_{macro}	**0.82**	0.81	0.75	0.67	0.73	0.66
$Prec_{micro}$	**0.95**	0.94	0.89	0.81	0.87	0.80
Rec_{micro}	**0.77**	0.76	0.73	0.68	0.72	0.68
F^1_{micro}	**0.85**	0.84	0.80	0.74	0.79	0.74
$Prec_{weight}$	**0.95**	0.94	0.89	0.81	0.87	0.80
Rec_{weight}	**0.77**	0.76	0.73	0.68	0.72	0.68
F^1_{weight}	**0.85**	0.84	0.80	0.74	0.78	0.73
$Prec_{inst}$	**0.95**	0.94	0.89	0.82	0.87	0.81
Rec_{inst}	**0.84**	0.84	0.79	0.74	0.78	0.74
F^1_{inst}	**0.88**	0.87	0.82	0.76	0.81	0.75

TABLE III
MULTI-LABEL PERFORMANCE METRICS

urations. Consistent with previous analysis, we observe better performance with the custom CNN and MC Dropout BCNN model. Table V presents the same information for multi-class classification and demonstrates the same broad trends discussed below.

In general, models perform the best on Classes D and E and worse on Classes A, B, and C. Intuitively, given that Class E represents the ”no ship” class and Class D encompasses the largest targets, this might not be a surprising result. Classes A, B, and C are more challenging for the models to distinguish and perform well on. We speculate that the main reason behind this could be the similarity of some target signatures across the three classes and large intra-class signature variation (see the ontology in Table I).

Fig. 5 illustrates a key benefit of the use of BCNNs as compared to traditional CNNs: the ability to get uncertainty measurements on each prediction. Not only does this value give us more insight into the performance of our model, it can also be used to triage only the most ambiguous classifications for further analysis by experts [36]. Examining the multi-class MC Dropout model in Fig. 5 reveals that filtering out all samples with H_p^* greater than 0.375 retains about 80% of the

	Custom						ResNet32v1					
	Dropout			Flipout			Dropout			Flipout		
Class	$Prec$	Rec	F^1	$Prec$	Rec	F^1	$Prec$	Rec	F^1	$Prec$	Rec	F^1
A	0.95	0.74	0.83	0.91	0.67	0.77	0.86	0.66	0.75	0.77	0.62	0.69
B	0.94	0.61	0.74	0.83	0.53	0.65	0.81	0.46	0.59	0.64	0.4	0.49
C	0.94	0.58	0.72	0.89	0.52	0.66	0.88	0.51	0.64	0.73	0.49	0.59
D	0.94	0.79	0.86	0.91	0.78	0.84	0.88	0.79	0.84	0.85	0.76	0.80
E	0.92	0.88	0.90	0.84	0.84	0.84	0.83	0.80	0.82	0.77	0.73	0.75

TABLE IV
MULTI-LABEL PER-CLASS METRICS

	Custom						ResNet32v1					
	Dropout			Flipout			Dropout			Flipout		
Class	$Prec$	Rec	F^1	$Prec$	Rec	F^1	$Prec$	Rec	F^1	$Prec$	Rec	F^1
A	0.71	0.55	0.62	0.68	0.49	0.57	0.55	0.48	0.51	0.55	0.38	0.45
B	0.70	0.59	0.64	0.60	0.55	0.58	0.49	0.48	0.49	0.51	0.45	0.48
C	0.75	0.77	0.76	0.74	0.71	0.72	0.72	0.61	0.66	0.68	0.57	0.62
D	0.80	0.85	0.83	0.77	0.84	0.80	0.73	0.81	0.77	0.69	0.81	0.74
E	0.94	0.95	0.94	0.91	0.92	0.91	0.89	0.87	0.88	0.85	0.82	0.84

TABLE V
MULTI-CLASS PER-CLASS METRICS

samples but improves the weighted F^1 score from 0.82 to 0.90. For the multi-label MC Dropout model, using the same filter results in retaining 86% of the data and increases the weighted F^1 score from 0.84 to 0.88. Thus the model is able to achieve significantly higher performance on a relatively large portion of the original dataset by removing the samples it is most uncertain about. These samples can then be put in a queue for further expert analysis. In crowded hydroacoustic environments, prioritizing samples by their uncertainty for analysis by sonar operators can enable ships, submarines, or shore monitoring stations to more efficiently process and categorize sonar contacts and more effectively allocate the scarce resources of operator time and attention.

Mean predictive probability, Eq. 6, is calculated through Monte Carlo integration with T samples. We evaluate the consistency of the prediction relative to the number of samples for a custom model in both dropout and flipout configuration, Table II. It can be observed that in both

Bayesian configurations the weighted F^1 score becomes consistent within T=10 Monte Carlo samples for flipout and T=20 Monte Carlo samples for dropout, see Fig.4.

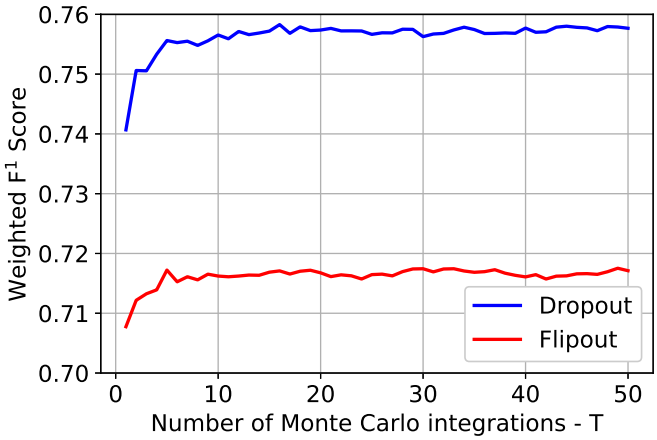


Fig. 4. Consistency of the prediction, in terms of weighted F^1 score, as a function of Monte Carlo sample size T.

In this work we chose to use T=50 given that it provided consistent predictive probability and that it was comparable to T sizes reported in the literature. For example, Filos et al. [36] used T=100 when comparing dropout and flipout methods in medical applications.

We recognize that the number of Monte Carlo samples required to achieve consistency in prediction is affected by numerous hyperparameters, including the underlying data distribution, model architecture and the choice of the Bayesian method (e.g. flipout, dropout, reparametrization), to name some. In practice, especially in live inference applications, one will likely choose to utilize a smaller sample size after conducting similar analysis to the one above. This involves the likely trade-off between computational and/or power requirements on one hand and consistency of prediction and/or calibration of uncertainty on the other.

Both MC Dropout and flipout Bayesian architectures produce calibrated uncertainties, see Fig.5. However, our overall results indicate better performance for MC Dropout model architectures across the evaluated metrics for both multi-class and multi-label classification tasks as shown in Table II and Table III. Given that the MC Dropout models have significantly fewer parameters and take less computational time than their flipout counterparts, MC Dropout is a promising option for sonar classification, especially for use on unmanned vehicles and other embedded systems. By producing calibrated uncertainties we also confirm that the used

uncertainty formulation, Eq. 8, suffices for our purposes in spite of additional uncertainty from approximation errors (e.g. Monte Carlo gradient estimation).

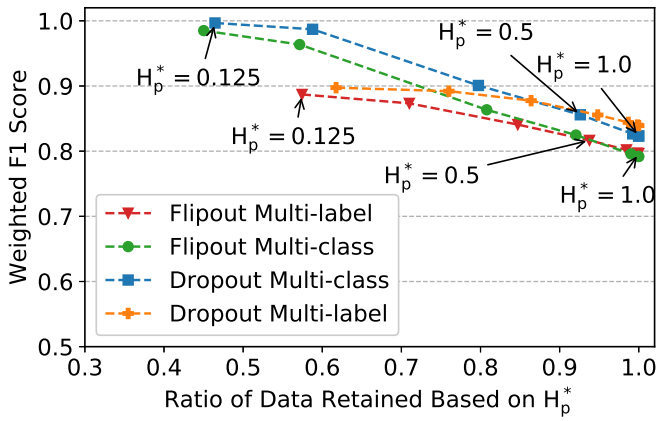


Fig. 5. Each point represents the the model's weighted average F^1 score vs. the ratio of overall number of samples retained when filtering out all samples above a certain predictive uncertainty value (measured by H_p^*)

C. Case Studies and Analysis

To explore how all of these results fit together in practice, we selected a ship and examined in detail the predictions our model made as that ship transited past the HARP sensor, which we have called Case Study I. The ship selected was a car carrier which sailed near the HARP sensor over about a four-hour period on February 25, 2013. We used the corresponding AIS information to build a range versus time plot and examined the model's predictive output and uncertainty along the track as shown in Fig. 6. As the ship approached, the model made several erroneous predictions and had high uncertainty until a range of roughly 15 km. With decreasing range, more sound information began to reach the sensor, making the predictions both more accurate and more certain. As the ship passes its closest point of approach and begins increasing range, we see the same effect, with less accurate and less certain predictions farther from the sensor.

The model predictions and uncertainties also capture two other hydroacoustic phenomena. The first is the bow null effect acoustic shadow zone, which occurs when the radiated engine and propeller noise (most often the main source of ship noise and generated towards the aft end of the ship) is partially blocked by the hull of an approaching ship [56]. When the ship is moving away, the stern is exposed and the noise reaches the sensor unimpeded. This is seen most clearly

in the large uncertainties and missed predictions on the approaching track, which continue in to a range of about 12 km. In contrast, on opening range, the uncertainties do not consistently rise until roughly 17 km, which is also where we observe the first missed prediction. Uncertainties also show a small spike as the ship is very close to the sensor, revealing the second phenomena: interference and acoustic bleed-over caused by the high sound pressure levels reaching the sensor. The observation of these two phenomena, as well as better performance at closer ranges, in Case Study I matches what would be expected in real-world conditions and demonstrates the model's ability to reflect reality in its performance and uncertainty measurements.

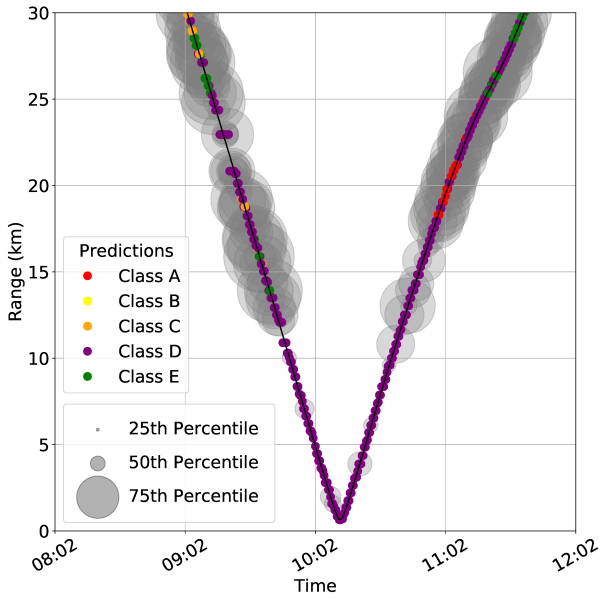


Fig. 6. Case Study I - February 25 2013, Car Carrier; range vs. time plot of a known target AIS track overlaid with BCNN classification output and predictive entropy, H_p. Classification outputs are color coded relative to the class where Car Carrier (classD) is correctly classified with magenta. H_p is scaled such that the size of the gray transparent circle corresponds to the percentile of the value range of H_p, with smaller circles corresponding to lower H_p values, and thus higher certainty. Predictions and H_p are from the custom multi-class MC Dropout BCNN trained on the full HARP dataset (see third column, Custom/Drop, in Table II).

As another case study, we trained additional versions of our custom model on a seasonal subset of the data. We preserve training strategy reported in Section III-C. In Case Study II, we looked at the performance of our general multi-class models trained on the whole year's data (see Table II) compared to the performance of models trained only on data from the winter months (December to March). The results of cross-testing both sets of models on the large and

small dataset are summarized in Table VI. While the models trained only on the winter data had excellent performance on a held-out test set from the winter, their predictive power was not generalizable, with poor performance on the full year's data. The models trained on the whole dataset, in contrast, were unable to reach the peak performance of those trained on the smaller dataset, but are able to perform much better on the whole range of data. They also lose only a small amount of their overall performance when making predictions on the smaller dataset.

These results demonstrate the effects of seasonal variation on model performance as well as the diverse set of underlying distributions required in order to train a generalizable classifier for underwater soundscapes. As seen in Fig. 2, seasonal changes to the water column's SSP affects how the noise from even the same ship on the same track is received by the sensor depending on the time of year. Datasets which are based on samples collected over a relatively brief period are likely to be biased. This decreases the practical utility of models trained on them, even if the models seem to have solid performance. In order to capture all of these differences, datasets with samples from all throughout the year, and ideally across multiple years, are needed. Another approach could be to train multiple models with data from different years but the same season, thus creating several "seasonal expert" classifiers.

Multi-Class	Trained on:	Small			Large		
Performance on:		Det	Drop	Flip	Det	Drop	Flip
Small	Acc	0.922	0.937	0.917	0.798	0.780	0.782
	$Prec_{weight}$	0.92	0.94	0.92	0.80	0.78	0.78
	Rec_{weight}	0.92	0.94	0.92	0.80	0.78	0.78
	F^1_{weight}	0.92	0.94	0.92	0.79	0.77	0.77
Large	Acc	0.482	0.481	0.472	0.807	0.826	0.796
	$Prec_{weight}$	0.51	0.50	0.49	0.80	0.82	0.79
	Rec_{weight}	0.48	0.48	0.47	0.81	0.83	0.80
	F^1_{weight}	0.44	0.44	0.44	0.80	0.82	0.79

TABLE VI

CASE STUDY II - CROSS-COMPARISON OF THE MULTI-CLASS PERFORMANCE OF MODELS TRAINED ON THE FULL DATASET VS. MODELS TRAINED ON A SEASONAL SUBSET. ALL MODELS ARE BASED ON OUR CUSTOM ARCHITECTURE. THE LARGE DATASET CONSISTS OF SAMPLES FROM A FULL YEAR, FROM DECEMBER 2012 TO NOVEMBER 2013. THE SMALL DATASET IS A SUBSET OF THE LARGER, CONSISTING ONLY OF THE SAMPLES FROM DECEMBER 2012 TO MARCH 2013.

V. Conclusion

The classification of underwater sounds is of great interest to the community and has seen significant progress with the proliferation of deep learning. This work addressed several challenges of using deep learning models for classification in acoustically heterogeneous environments and effectively establishes benchmark performance for both the multi-label and multi-class classification of underwater soundscapes. Additionally, this is also a first study demonstrating the quality and the utility of the uncertainty of neural network classification with a ship-based ontology on underwater soundscapes. Our best performing Bayesian model developed for the multi-label task achieves a weighted F^1 score of 0.84 and the model developed on the multi-class task achieves a weighted F^1 score of 0.82. In both of those tasks, models simultaneously offered measurement of uncertainty in per sample classification. This was achieved by adopting Bayesian Deep Learning, a new and emerging field, which can have significant implications on the proliferation of deep learning models in production. The presented analysis is universal and applicable to any other classification or regression task in soundscape monitoring. The demonstrated results and analysis of uncertainty in the first case study correlated well with a physical understanding of sound propagation. Moreover, in the second case study we demonstrated the ability to preserve model performance with the seasonal variation of underlying sound speed profiles, which was enabled by training on one of the largest datasets used in this type of analysis. Overall, the proposed approaches can have significant impact on autonomous monitoring of ocean resources through passive sonar.

References

[1] D. Gillespie, "Detection and classification of right whale calls using an 'edge' detector operating on a smoothed spectrogram," *Canadian Acoustics*, vol. 32, no. 2, pp. 39–47, Jun. 2004.

[2] C. McQuay, F. Sattar, and P. F. Driessen, "Deep learning for hydrophone big data," in *2017 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*. Victoria, BC: IEEE, Aug. 2017, pp. 1–6.

[3] M. Thomas, B. Martin, K. Kowarski, B. Gaudet, and S. Matwin, "Marine Mammal Species Classification using Convolutional Neural Networks and a Novel Acoustic Representation," in *Machine Learning and Knowledge Discovery in Databases*, Würzburg, Germany, Jul. 2019.

[4] A. Tesei, R. Been, and F. Meyer, "Continuous real-time acoustic surveillance of fast surface vessels," in *UACE 2017 Proceedings*, Skiathos, Greece, 2017, pp. 463–468.

[5] H. Berg, K. T. Hjelmervik, D. H. S. Stender, and T. S. Sastad, "A comparison of different machine learning algorithms for automatic classification of sonar targets," in *OCEANS 2016 MTS/IEEE Monterey*. Monterey, CA, USA: IEEE, Sep. 2016, pp. 1–8.

[6] D. Neupane and J. Seok, "A Review on Deep Learning-Based Approaches for Automatic Sonar Target Recognition," *Electronics*, vol. 9, no. 11, p. 1972, Nov. 2020, number: 11 Publisher: Multidisciplinary Digital Publishing Institute.

[7] M. J. Bianco, P. Gerstoft, J. Traer, E. Ozanich, M. A. Roch, S. Gannot, and C.-A. Deledalle, "Machine learning in acoustics: Theory and applications," *The Journal of the Acoustical Society of America*, vol. 146, no. 5, pp. 3590–3628, Nov. 2019.

[8] A. Zak, "Ship's Hydroacoustics Signatures Classification Using Neural Networks," in *Self Organizing Maps - Applications and Novel Algorithm Design*, J. I. Mwasiagi, Ed. InTech, Jan. 2011.

[9] D. Santos-Domínguez, S. Torres-Guijarro, A. Cardenal-López, and A. Pena-Gimenez, "ShipsEar: An underwater vessel noise database," *Applied Acoustics*, vol. 113, pp. 64–69, Dec. 2016.

[10] H. Niu, E. Ozanich, and P. Gerstoft, "Ship localization in Santa Barbara Channel using machine learning classifiers," *The Journal of the Acoustical Society of America*, vol. 142, no. 5, pp. EL455–EL460, Nov. 2017.

[11] T. B. Neilsen, C. D. Escobar-Amado, M. C. Acree, W. S. Hodgkiss, D. F. Van Komen, D. P. Knobles, M. Badiey, and J. Castro-Correa, "Learning location and seabed type from a moving mid-frequency source," *The Journal of the Acoustical Society of America*, vol. 149, no. 1, pp. 692–705, Jan. 2021.

[12] A. M. Pfau, "Multi-label Classification of Underwater Soundscapes Using Deep Convolutional Neural Networks," Master's thesis, Naval Postgraduate School, Monterey, CA, 2020. [Online]. Available: https://calhoun.nps.edu/handle/10945/66705

[13] M. F. McKenna, D. Ross, S. M. Wiggins, and J. A. Hildebrand, "Underwater radiated noise from modern commercial ships," *The Journal of the Acoustical Society of America*, vol. 131, no. 1, pp. 92–103, Jan. 2012.

[14] P. T. Arveson and D. J. Vendittis, "Radiated noise characteristics of a modern cargo ship," *The Journal of the Acoustical Society of America*, vol. 107, no. 1, pp. 118–129, Jan. 2000.

[15] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. Wilson, "CNN Architectures for Large-Scale Audio Classification," in *ICASSP 2017*. New Orleans, LA: IEEE, Jan. 2017.

[16] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*. PMLR, 2016, pp. 1050–1059.

[17] H. Wang and D.-Y. Yeung, "A Survey on Bayesian Deep Learning," *ACM Computing Surveys*, vol. 53, no. 5, pp. 1–37, Oct. 2020.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[19] ——, "Identity mappings in deep residual networks," in *European conference on computer vision*. Springer, 2016, pp. 630–645.

[20] S. M. Wiggins and J. A. Hildebrand, "High-frequency Acoustic Recording Package (HARP) for broad-band, long-term marine mammal monitoring," in *2007 Symposium on Underwater Technology and Workshop on Scientific Use of Submarine Cables and Related Technologies*. Tokyo, Japan: IEEE, Apr. 2007, pp. 551–557.

[21] VesselFinder website available at https://www.vesselfinder.com/ (Last viewed Jun 29, 2021).

[22] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S.-Y. Chang, and T. Sainath, "Deep learning for audio signal processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 206–219, 2019.

[23] G.-D. Wu and C.-T. Lin, "Word boundary detection with mel-scale frequency bank in noisy environment," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 5, pp. 541–554, 2000.

[24] R. Allen Jr, "Naval oceanographic office global gridded physical profile data from the us navy's generalized digital environmental model (gdem) product database (nodc accession 9600094)."

[25] K. V. Mackenzie, "Nine-term equation for sound speed in the oceans," *The Journal of the Acoustical Society of America*, vol. 70, no. 3, pp. 807–812, 1981.

[26] K. Shridhar, F. Laumann, and M. Liwicki, "A comprehensive guide to bayesian convolutional neural network with variational inference," *arXiv preprint arXiv:1901.02731*, 2019.

[27] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[28] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *International Conference on Machine Learning*. PMLR, 2015, pp. 1613–1622.

[29] A. Graves, "Practical variational inference for neural networks," *Advances in neural information processing systems*, vol. 24, 2011.

[30] D. P. Kingma, T. Salimans, and M. Welling, "Variational dropout and the local reparameterization trick," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015.

[31] Y. Wen, P. Vicol, J. Ba, D. Tran, and R. Grosse, "Flipout: Efficient pseudo-independent weight perturbations on mini-batches," in *International Conference on Learning Representations*, 2018.

[32] R. Feng, N. Balling, D. Grana, J. S. Dramsch, and T. M. Hansen, "Bayesian convolutional neural networks for seismic facies classification," *IEEE Transactions on Geoscience and Remote Sensing*, 2021.

[33] S. Mohamed, M. Rosca, M. Figurnov, and A. Mnih, "Monte carlo gradient estimation in machine learning." *J. Mach. Learn. Res.*, vol. 21, no. 132, pp. 1–62, 2020.

[34] J. Paisley, D. Blei, and M. Jordan, "Variational bayesian inference with stochastic search," *arXiv preprint arXiv:1206.6430*, 2012.

[35] J. V. Dillon, I. Langmore, D. Tran, E. Brevdo, S. Vasudevan, D. Moore, B. Patton, A. Alemi, M. Hoffman, and R. A. Saurous, "Tensorflow distributions," *arXiv preprint arXiv:1711.10604*, 2017.

[36] A. Filos, S. Farquhar, A. N. Gomez, T. G. Rudner, Z. Kenton, L. Smith, M. Alizadeh, A. de Kroon, and Y. Gal, "A systematic comparison of bayesian deep learning robustness in diabetic retinopathy tasks," *arXiv preprint arXiv:1912.10481*, 2019.

[37] Y. Gal, R. Islam, and Z. Ghahramani, "Deep bayesian active learning with image data," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1183–1192.

[38] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[39] Y. Gal, J. Hron, and A. Kendall, "Concrete dropout," *arXiv preprint arXiv:1705.07832*, 2017.

[40] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*, ser. Adaptive computation and machine learning. Cambridge, Massachusetts London, England: The MIT Press, 2016.

[41] Z. Nado, N. Band, M. Collier, J. Djolonga, M. Dusenberry, S. Farquhar, A. Filos, M. Havasi, R. Jenatton, G. Jerfel, J. Liu, Z. Mariet, J. Nixon, S. Padhy, J. Ren, T. Rudner, Y. Wen, F. Wenzel, K. Murphy, D. Sculley, B. Lakshminarayanan, J. Snoek, Y. Gal, and D. Tran, "Uncertainty Baselines: Benchmarks for uncertainty & robustness in deep learning," *arXiv preprint arXiv:2106.04015*, 2021.

[42] L. A. F. Park and S. Simoff, "Using Entropy as a Measure of Acceptance for Multi-label Classification," in *Advances in Intelligent Data Analysis XIV*, E. Fromont, T. De Bie, and M. van Leeuwen, Eds. Cham: Springer International Publishing, 2015, pp. 217–228.

[43] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[44] H. Guo, K. Zheng, X. Fan, H. Yu, and S. Wang, "Visual attention consistency under image transforms for multi-label image classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 729–739.

[45] T. Durand, N. Mehrasa, and G. Mori, "Learning a deep convnet for multi-label classification with partial labels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[46] J. Wehrmann, R. Cerri, and R. Barros, "Hierarchical multi-label classification networks," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5075–5084.

[47] D. Tran, M. Dusenberry, M. van der Wilk, and D. Hafner, "Bayesian layers: A module for neural network uncertainty," in *Advances in Neural Information Processing Systems*, 2019, pp. 14 660–14 672.

[48] J. Pons and X. Serra, "Designing efficient architectures for modeling temporal features with convolutional neural networks," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. New Orleans, LA: IEEE, Mar. 2017, pp. 2472–2476.

[49] R. Mars, P. Pratik, S. Nagisetty, and C. Lim, "Acoustic Scene Classification from Binaural Signals using Convolutional Neural Networks," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*. New York University, 2019, pp. 149–153.

[50] B. M. Ozyildirim and S. Kartal, "Comparison of Deep Convolutional Neural Network Structures The effect of layer counts and kernel sizes," in *Fourth International Conference on Advances in Information Processing and Communication Technology - IPCT 2016*. Institute of Research Engineers and Doctors, Aug. 2016, pp. 16–19.

[51] Y. Li, C. Wei, and T. Ma, "Towards explaining the regularization effect of initial large learning rate in training neural networks," in *Advances in Neural Information Processing Systems*, 2019, pp. 11 674–11 685.

[52] M. Grandini, E. Bagli, and G. Visani, "Metrics for Multi-Class Classification: an Overview," *arXiv*, Aug. 2020.

[53] M.-L. Zhang and Z.-H. Zhou, "A Review on Multi-Label Learning Algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1819–1837, Aug. 2014.

[54] S. Godbole and S. Sarawagi, "Discriminative Methods for Multi-labeled Classification," in *Advances in Knowledge Discovery and Data Mining*, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, H. Dai, R. Srikant, and C. Zhang, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, vol. 3056, pp. 22–30.

[55] G. Tsoumakas and I. Katakis, "Multi-Label Classification: An Overview," *International Journal of Data Warehousing and Mining*, vol. 3, no. 3, pp. 1–13, Jul. 2007.

[56] J. K. Allen, M. L. Peterson, G. V. Sharrard, D. L. Wright, and S. K. Todd, "Radiated noise from commercial ships in the Gulf of Maine: Implications for whale/vessel collisions," *The Journal of the Acoustical Society of America*, vol. 132, no. 3, pp. EL229–EL235, Sep. 2012.