2021-09

ENHANCED MULTI-LABEL CLASSIFICATION OF HETEROGENEOUS UNDERWATER SOUNDSCAPES BY CONVOLUTIONAL NEURAL NETWORKS USING BAYESIAN DEEP LEARNING

Beckler, Brandon M.

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/70790

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

ENHANCED MULTI-LABEL CLASSIFICATION OF HETEROGENEOUS UNDERWATER SOUNDSCAPES BY CONVOLUTIONAL NEURAL NETWORKS USING BAYESIAN DEEP LEARNING

by

Brandon M. Beckler

September 2021

Thesis Advisor:	Marko Orescanin
Second Reader:	Vinnie Monaco

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** September 2021	**3. REPORT TYPE AND DATES COVERED** Master’s thesis

4. TITLE AND SUBTITLE
ENHANCED MULTI-LABEL CLASSIFICATION OF HETEROGENEOUS UNDERWATER SOUNDSCAPES BY CONVOLUTIONAL NEURAL NETWORKS USING BAYESIAN DEEP LEARNING

5. FUNDING NUMBERS

6. AUTHOR(S) Brandon M. Beckler

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release. Distribution is unlimited.

12b. DISTRIBUTION CODE
A

13. ABSTRACT (maximum 200 words)

The classification of underwater soundscapes is a challenging task for humans as well as machine learning systems. This is largely due to the heterogenous nature of these soundscapes, especially in coastal zones close to human settlements, where multiple ships and other man-made and natural sound sources are often present simultaneously. This thesis proposes a Bayesian deep learning approach that can accurately classify multiple ships simultaneously present in the vicinity of a sensor (multi-label classification) while also providing an uncertainty measurement for 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. The best performing Bayesian architecture on the multi-label task achieves a weighted F1 score of 0.84, where each prediction is accompanied by a measurement of uncertainty that is used to further enhance the understanding of model predictions. Ships, submarines, and unmanned underwater vehicles can use this classification system to aid in the identification, tracking, and/or targeting of contacts to help maintain safety of navigation, to aid in the real-time interdiction of illicit activities (such as drug or human smuggling and covert vessel transits), and to provide port security monitoring while uncertainty filters can help sonar operators prioritize contacts for further analysis.

14. SUBJECT TERMS
machine learning, convolutional neural networks, Bayesian deep learning, soundscape classification, underwater soundscape, sonar

15. NUMBER OF PAGES
67

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

THIS PAGE INTENTIONALLY LEFT BLANK

ENHANCED MULTI-LABEL CLASSIFICATION OF HETEROGENEOUS UNDERWATER SOUNDSCAPES BY CONVOLUTIONAL NEURAL NETWORKS USING BAYESIAN DEEP LEARNING

Brandon M. Beckler
Lieutenant Commander, United States Navy
BS, U.S. Naval Academy, 2011

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
September 2021

Approved by: Marko Orescanin
Advisor

Vinnie Monaco
Second Reader

Gurminder Singh
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The classification of underwater soundscapes is a challenging task for humans as well as machine learning systems. This is largely due to the heterogenous nature of these soundscapes, especially in coastal zones close to human settlements, where multiple ships and other man-made and natural sound sources are often present simultaneously. This thesis proposes a Bayesian deep learning approach that can accurately classify multiple ships simultaneously present in the vicinity of a sensor (multi-label classification) while also providing an uncertainty measurement for 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. The best performing Bayesian architecture on the multi-label task achieves a weighted F1 score of 0.84, where each prediction is accompanied by a measurement of uncertainty that is used to further enhance the understanding of model predictions. Ships, submarines, and unmanned underwater vehicles can use this classification system to aid in the identification, tracking, and/or targeting of contacts to help maintain safety of navigation, to aid in the real-time interdiction of illicit activities (such as drug or human smuggling and covert vessel transits), and to provide port security monitoring while uncertainty filters can help sonar operators prioritize contacts for further analysis.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

1 Introduction 1

2 Background and Related Work 5

2.1 Neural Networks and Deep Learning 5

2.2 Underwater Soundscapes 9

2.3 Bayesian Deep Learning 10

3 Methodology 15

3.1 Dataset 15

3.2 Environment 18

3.3 Architecture 21

3.4 Evaluation Metrics. 23

3.5 Experiments 27

4 Results 29

4.1 Overall Results 29

4.2 Uncertainty Filtering 31

4.3 Case Study I 34

4.4 Case Study II 37

5 Conclusion 39

5.1 Future Work 40

List of References 41

Initial Distribution List 49

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

Figure 2.1 Softmax Output Layer 6

Figure 2.2 Neural Network Architecture 7

Figure 2.3 Convolutional Layers 8

Figure 2.4 Convolutional Neural Network Architecture 9

Figure 3.1 Generic Seasonal Sound Speed Variation 19

Figure 3.2 Monthly Sound Speed Profiles 20

Figure 3.3 Model Architecture 23

Figure 4.1 F^1 Filtered by Entropy 32

Figure 4.2 Histogram of Normalized Entropy 33

Figure 4.3 Case Study I Time-Range Plot 35

Figure 4.4 Case Study I Geographic Plot 36

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

Table 3.1 Ship Classes 16

Table 3.2 Label Cardinality and Label Density 17

Table 4.1 Multi-class Performance Metrics 29

Table 4.2 Multi-label Performance Metrics 31

Table 4.3 Case Study II 37

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

AI	artificial intelligence
AIS	automatic identification system
BNN	Bayesian neural network
BDL	Bayesian deep learning
CNN	convolutional neural network
DOD	Department of Defense
GDEM	Generalized Digital Environmental Model
HARP	High-frequency Acoustic Recording Package
IMO	International Maritime Organization
JAIC	Joint Artificial Intelligence Center
KL	Kullback-Leibler
ML	machine learning
MMSI	Maritime Mobile Service Identity
NPS	Naval Postgraduate School
ReLU	Rectified Linear Unit
ResNet	Residual Network
ROC	receiver operating characteristic
SSP	sound speed profile
STFT	Short Time Fourier Transform
UUV	unmanned underwater vehicle
VI	variational inference

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgments

I would like to thank my thesis advisor, Dr. Marko Orescanin, for his guidance, mentoring, and patience in aiding me through this process. I'd also like to thank Dr. Vinnie Monaco for his insightful comments on drafts of this work.

Special thanks to LT Andrew Pfau, USN, now instructing at the U.S. Naval Academy, whose NPS master's thesis began this line of work and who has continued to provide support and advice. Thanks also to several fellow NPS students who contributed in various ways: LT Sabrina Atchley, USN, who provided the Bayesian neural network code that enabled my experiments; Capt. Benjamin Marsh, USMC, who aided in understanding the math behind Bayesian techniques; and LT Nicholas Villemez, USN, who greatly assisted in data visualization.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1: Introduction

This chapter is adapted from [1], previously published by the Journal of Oceanic Engineering, ©2021 IEEE[1] [2]

The past decade has seen an increase in research into artificial intelligence (AI) and machine learning (ML) systems, along with a concomitant rise in the application of such systems, creating what some have called an AI "frenzy" [2]. Buoyed by the growth of big data and the Internet of things, AI/ML has begun to affect nearly every part of society. These systems have the potential to become "a revolution that will transform how we live, work, and think" [3].

"How we fight" could easily be added to this list, and indeed the U.S. Department of Defense (DOD) has recognized the need to be a leader in AI/ML. In 2018, the DOD formed the Joint Artificial Intelligence Center (JAIC) to "seize upon the transformative potential of Artificial Intelligence technology for the benefit of America's national security" and released an "AI Strategy" in 2019 [4]. One of the ways that the JAIC, and other DOD AI initiatives, will accomplish its mission will be to develop systems that aid human operators in sifting through the vast amounts of data (such as imagery, or network traffic) that the DOD collects every day. Effective AI systems could help operators to quickly and more accurately classify and interpret this data, ultimately enabling more rapid employment of the information in the operational arena.

[1] Reprinted, with permission, from Beckler et al., "Multi-Label Classification of Heterogeneous Underwater Soundscapes with Bayesian Deep Learning," *IEEE Journal of Oceanic Engineering*, MON 2021. This publication is a work of the U.S. government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States. IEEE will claim and protect its copyright in international jurisdictions where permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

[2] In reference to IEEE copyrighted material that is used with permission in this thesis, the IEEE does not endorse any of the Naval Postgraduate School's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink. If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

This thesis will specifically examine how AI/ML techniques can aid sonar operators in the processing and classification of sonar data. The ability to classify sonar signals quickly and accurately is vital to the conduct of undersea warfare. This is true no matter the platform, but it is especially so for submarines and unmanned underwater vehicles (UUVs). These platforms rely primarily (or almost exclusively when operating far from the surface) on sonar for information about the operational environment, including for targeting and threat warning.

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 [5]–[7]. Shipping noise can have adverse impacts on marine mammal populations and the measurement and modeling of shipping noise is important in predicting environmental impacts for conservation efforts. Such research aids autonomous monitoring of fisheries and fishery enforcement by government and environmental groups [8]. Ships, submarines, aircraft, and UUVs can use passive sonar classification systems to aid in the identification, tracking, and/or targeting of contacts, to help maintain safety of navigation, to aid in the real-time interdiction of illicit activities (such as drug or human smuggling and covert vessel transits), and to provide port security monitoring [9], [10].

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 (multi-class classification) [5], [6]. This approach, however, is an inaccurate representation of the heterogeneous underwater acoustic environment where multiple ship and other man-made signals, as well as biological and natural sound sources, are often simultaneously present. This is especially true of underwater soundscapes in coastal zones close to human settlements. Classification of such heterogeneous underwater soundscapes is a challenging task for humans as well as ML systems. ML 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 thesis addresses that challenge by introducing the multi-label classification task. 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 [11], this is achieved by expanding upon and evaluating a custom multi-label convolutional neural network (CNN) architecture which was first developed by Andrew Pfau in his previously published Naval Postgraduate School (NPS) master's thesis [12]. To address the lack of uncertainty measurement in the classification estimates, a Bayesian deep learning (BDL) approach is adopted borrowing techniques developed by Sabrina Atchley for her upcoming NPS master's thesis [13]. BDL combines deep learning techniques and Bayesian theory to enable models that provide uncertainty measurements and are more robust to overfitting relative to analogous deterministic (classical) neural network architectures [14]. Uncertainty measurements also allow for a deeper understanding of the model's predictions [15].

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, for both multi-class and multi-label classification tasks, to estimate predictive uncertainty. Both deterministic and Bayesian configurations of deep Residual Network (ResNet) model architectures [16], [17] and a custom CNN architecture are analyzed and benchmarked on these classification tasks. The uncertainty of predictions of ship classification is suggested as a distinctive improvement of BDL architectures over deterministic models for underwater soundscape classification applications. Additionally, with more than 4,000 unique ships and over 3,400 hours of labeled audio data, the large size of the dataset used enabled a study of the impact of the seasonal variation of sound speed profile (SSP) on the bias and quality of classifications of developed deep learning models. Finally, a use-case study examines the quality of the measured uncertainty and correlates the uncertainty of classification to distance from the sensor and the bow-stern orientation.

The rest of this thesis is organized as follows. Chapter 2 provides background information and related work on the topics covered in this paper, including the use of AI/ML techniques, such as neural networks, in Section 2.1, the specific application of ML techniques to underwater soundscape classification in Section 2.2, and BDL techniques in Section 2.3. Chapter 3 outlines the methodology used, beginning with the dataset in Section 3.1, the environment in Section 3.2, and the classification architecture in Section 3.3. Section 3.4

then describes the metrics used to evaluate the classification models, and finally, Section 3.5 details the experimental setup. Chapter 4 begins by examining the overall results of the experiments in Section 4.1, then moves to a more specific evaluation of the usefulness of the uncertainties produced by the BDL techniques in Section 4.2 before describing two real-world case studies in Sections 4.3 and 4.4. Chapter 5 provides concluding comments and some thoughts on possible future work in Section 5.1.

CHAPTER 2:
Background and Related Work

This chapter is adapted from [1], previously published by the Journal of Oceanic Engineering, ©2021 IEEE

2.1 Neural Networks and Deep Learning

While AI and ML are closely related, they are not exactly the same. AI is the science and engineering of making machines which exhibit intelligent behavior, as well as developing a computational understanding of those behaviors [18], [19]. ML is the study of methods that allow computers to "learn from data without being explicitly programmed" [20], [21], and these techniques are often the foundation of AI systems.

ML techniques have a surprisingly long history in computer science, given that they have only recently gained prominence. For example, the first artificial neural network, inspired by how neurons work in the brain, was proposed in 1943 [20]. Limitations in computer processing, data availability, training techniques, and theory drove ML research in other directions (such as decision trees, support vector machines, k-nearest neighbor, etc.) until the 1980s and 1990s [20]. Since then, advances in these areas have spurred rapid developments and significant performance improvements, creating neural networks that are powerful, versatile, and able to handle large, complex ML challenges [22]. Today, artificial neural networks and their variants are among the most popular ML techniques in many fields [18].

A basic neural network is made up of an input layer, a hidden layer and an output layer. The input layer has the same number of neurons as the number of input features (often plus a bias neuron) and simply passes its input to the neurons in the hidden layer across a number of connections. Each connection has a weight value assigned to it, and the neurons of the hidden layer compute a weighted sum of their input connections, and then pass this sum to some activation function. This activation function then computes an output for the hidden layer neuron, and then passes this as an input to the output layer neurons. The output layer has the same number of neurons as the properties which the network is trying to predict (such as classes for classification, or separate continuous variables for regression), and behavior similarly to the hidden layer neurons. The final output of this layer is then used, either directly or through another activation function, to make predictions [20], [23].

In multi-label classification, the sigmoid, or logistic, function is a common activation function which takes the inputs to a neuron, multiplies them by the weights, and outputs a number between zero and one which can be thought of as a probability [20]. These values from the output neurons are then compared against a threshold value; any class with an output at or above the threshold is labeled as present, while those with outputs below the threshold are labeled as not present. In multi-class classification, where the samples belong exclusively to one class, the output layer uses a softmax activation function, which ensures that the probability estimates for each class are values between zero and one and that the probabilities for all classes sum to one [20], see Figure 2.1 from Géron [20]. Also, see Section 3.3 for a discussion on how this thesis approaches these issues.

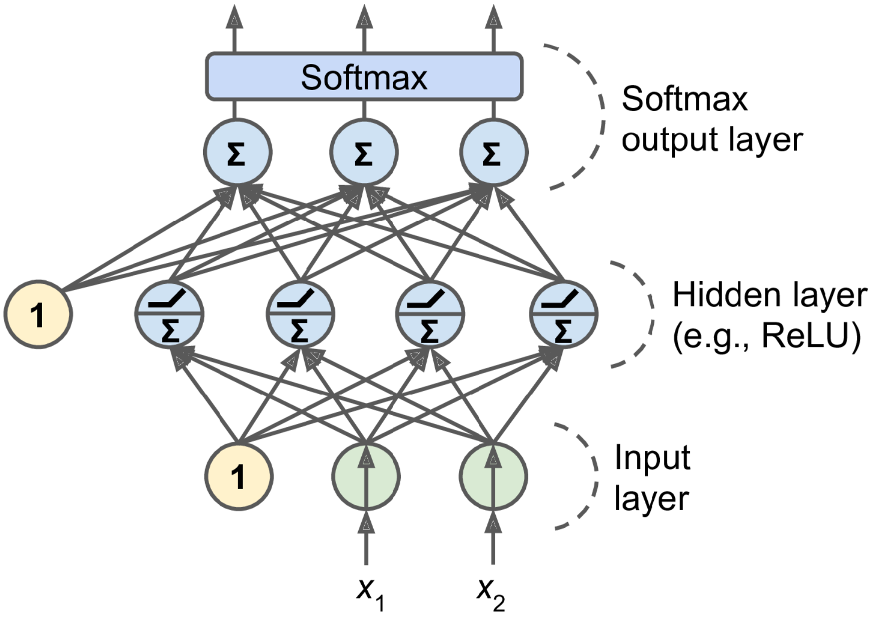


Figure 2.1. A neural network using a softmax output layer for multi-class classification. The final hidden layer can have any activation function (in this case, it is the Rectified Linear Unit (ReLU) activation function) and the softmax output layer will make all the class probabilities sum to one. The neurons with a "1" are bias neurons. Source: [20]

By passing training inputs through the network, measuring the prediction error, modifying the weights of the network in a way that minimizes that error, and iterating, the neural network learns the optimum weight values which enable it to achieve its peak performance. Figure 2.2, from Krizhevsky et al. [23], shows a simple neural network with one hidden layer.

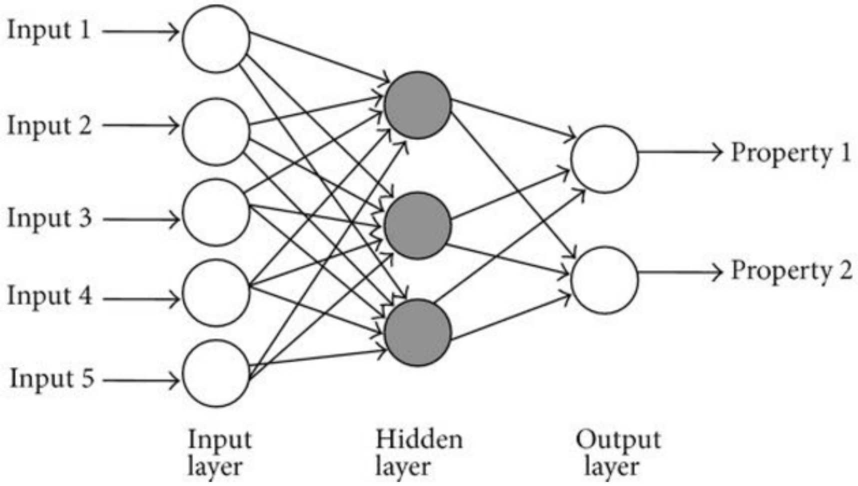


Figure 2.2. Generic example of the structure of a simple neural network. Source: [23].

As work on neural networks progressed, researchers found that they were able to see large performance gains and represent even more complex datasets by stacking more neural network layers on top of one another. These approaches became known as deep learning or deep neural networks, and, paired with large amounts of data available for training and fast computer processing, they became the standard for many ML tasks. Specifically, CNNs performed especially well on many pattern recognition tasks [3], [24].

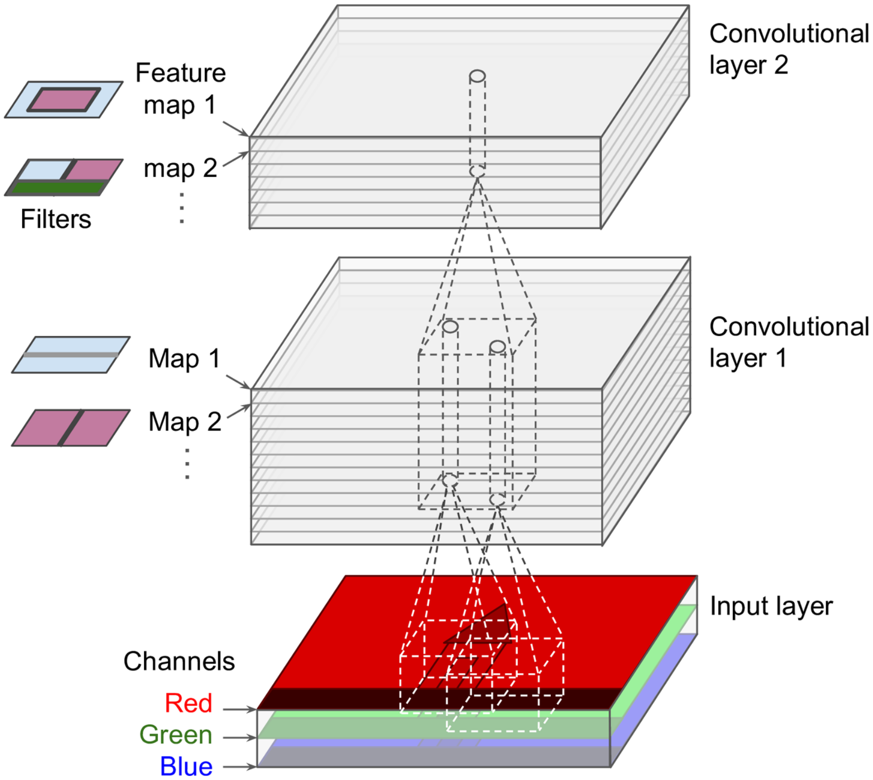


Figure 2.3. Convolutional layers with multiple feature maps applied to an image with three color channels. Source: [20].

CNNs use filters to connect subsets of inputs, known as local receptive fields, to one output neuron in a convolutional layer. This essentially compresses the information contained in the original inputs, significantly reducing the computations required in training, and allowing the CNN to learn general abstractions from the inputs. When all the neurons in a layer use the same filter on the inputs, the output is a feature map which highlights the regions in the input which most activated the filter [20]. By using multiple filters and stacking the feature maps, a CNN is able to detect the learned features anywhere in its input, making the CNN an excellent pattern detector, well-suited for tasks like image classification or object detection. Figure 2.3, from Géron [20], shows how convolutional layers with multiple filters can be applied to an image with red, green, and blue color channels.

These feature are then pooled into a sub-sample of the input, which reduces computational costs, memory requirements, and the number of parameters (which reduces model overfitting) [20]. Finally, one or more traditional, fully-connected neural network layers are added to the CNN to enable predictions as before. Figure 2.4, from Géron [20], presents a typical CNN architecture for image classification.

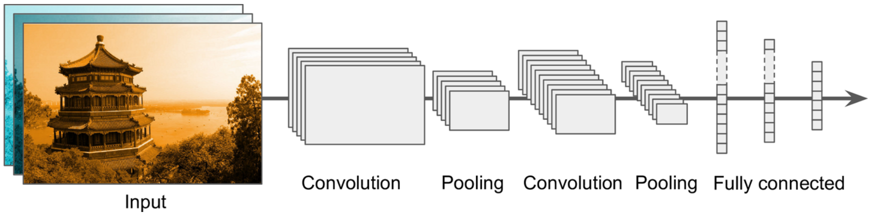


Figure 2.4. A typical CNN architecture. Source: [20].

While CNNs first gained popularity in image classification systems, they have recently begun to be applied to audio classification tasks as well [11]. These tasks have included speech recognition and localization, bioacoustics (classifying animal vocalizations), and the characterization of sound reverberation in natural environments [22]. Prior to the use of CNNs, other types of ML systems, such as support vector machines, were applied to audio classification tasks, but CNNs have proven more robust and have exceeded the performance of more traditional ML approaches [25].

2.2 Underwater Soundscapes

The use of ML algorithms for the classification of underwater sounds is well established [22]. Most research in this area, however, focuses on identification of biological sounds [5], [7] 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 [26]. Santos-Domínguez et al. report using only two hours of recordings [27], and Niu et al. use just three ships with 30 minutes of recording from each ship [28]. Berg et al. [9] and Neilsen et al. [29] 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 sailing on the ocean creates sound by operating different pieces of machinery, propeller cavitation, and the movement 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 [12]. The size, speed, and aspect to the sensor all affect the type and strength of signals received, as do oceanographic conditions such as temperature, salinity and pressure (primarily a function of depth) [30]. 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 [31]. McKenna et al. examined recordings of multiple commercial ships which show that the sound from container ships predominately falls below 40 Hertz and that all ships showed asymmetry in their signatures, with bow aspect radiated noise lower than stern aspect [30]. 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.

2.3 Bayesian Deep Learning

In statistical inference, Bayes' theorem allows the prior, a probability distribution that reflects preexisting beliefs about the relationships between data and latent variables, to be updated after the observation of more data [20]. Predictive models based on Bayes' theorem can be powerful tools, but since they are based on inferences from data, they must be able to handle uncertainty [32]. BDL combines this ability of Bayesian probabilistic models to provide uncertainty in predictions with the ability of neural networks to recognize patterns and relationships [15]. Specifically, model uncertainty is measured by placing a prior probability distribution over the model's weights in order to construct a Bayesian neural network (BNN) [33]. 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 $\hat{y}_n = f(\boldsymbol{x_n})$. The Bayesian approach assumes a prior distribution over the space of functions (i.e., a distribution over network parameters) $p(f)$ [34], the object being to quantify the posterior uncertainty over the network parameters $p(f \mid \mathcal{D})$ given a dataset $\mathcal{D}$. 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 f [35]:

$$p\left(\hat{y}|\boldsymbol{x}^*, \mathcal{D}\right) = \int_f p\left(\hat{y}|\boldsymbol{x}^*, f\right) p\left(f|\mathcal{D}\right) \tag{2.1}$$

In practice, because inference defined in Equation 2.1 is intractable due to calculation of the probability distribution $p(f \mid \mathcal{D})$, an approximate inference is used. This work evaluates variational inference (VI) approaches that approximate the posterior distribution $p(f \mid \mathcal{D}) \propto p(f)p(\mathcal{D} \mid f)$ by fitting an approximation $q_\theta(f) \approx p(f \mid \mathcal{D})$, where θ are the parameters of the probability distribution over weights [36]. In particular, loss is defined as a negative evidence lower bound function (commonly known as ELBO) and is minimized relative to θ [35], [36]:

$$\begin{aligned} \mathcal{L}(\theta) = &-\mathbb{E}_{q_\theta}\big[\log p\big(\mathcal{D} \mid f\big)\big] \\ &+ \mathrm{KL}\big(q_\theta(f) \mid\mid p(f)\big) \end{aligned} \tag{2.2}$$

where the first term represents the expected likelihood, which "describes how the variational distributions of the neural parameters explain the observed data" [37], and the second term is the Kullback-Leibler (KL) divergence measuring proximity between the posterior and prior densities [38].

Prediction uncertainty is induced by the uncertainty in weights and can be calculated by marginalizing over the approximate posterior using Monte Carlo integration [14] with T samples to calculate mean predictive probability:

$$
\begin{aligned}
p(\hat{y}=c \mid \boldsymbol{x}^*, \mathcal{D}) &= \int p(\hat{y}=c \mid \boldsymbol{x}^*, f)p(f \mid \mathcal{D})df \\
&\approx p(\hat{y}=c \mid, f)q_\theta(f)df \\
&\approx \frac{1}{T}\sum_{t=1}^{T} p(\hat{y}=c \mid \boldsymbol{x}^*, \hat{f}_t) \\
&\approx \frac{1}{T}\sum_{t=1}^{T} \hat{p}_{c_t} = \bar{p}_c
\end{aligned}
\tag{2.3}
$$

where $\hat{f}_t \sim q_\theta(f)$ and c represents all possible classes. This research evaluates fully factorized Gaussian posterior q_θ (and prior) with flipout Monte Carlo estimators of KL-divergence [36], [39]. Flipout is a method used to decorrelate gradients within a training mini-batch by implicitly sampling the weights of a neural network at training time in a stochastic, pseudo-independent manner [36]. Furthermore, final classification is assigned based on Equation 2.3. This assigns a class to each instance based on the greatest mean predictive probability.

Additionally, a simpler method than explicitly modeling distributions over weights, called Monte Carlo dropout, is evaluated [14]. Gal and Gharmanani [14] have shown that introducing dropout layers in inference, not just during training, is equivalent to Bernoulli approximation of the posterior q_θ over weights. The advantage of such an approach is that the number of neural network parameters is significantly smaller than that required for flipout or other VI techniques. This method also requires relatively minor changes to the neural network architectures and training processes used by traditional deterministic models.

Input mel-log spectrograms were classified from Equation 2.3 using argmax $\bar{p}_c$. Predictive entropy and total variance are used to quantify the uncertainty of BNN models [14], [40]. Predictive entropy measures the average amount of information encompassed by the predictive distribution and is a commonly used uncertainty metric [35]. It is given by:

$$H_p(\hat{y} \mid \boldsymbol{x}^*) = -\sum_c \bar{p}_c \log \bar{p}_c \tag{2.4}$$

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) [41] as shown in Equation 2.5.

$$H_p^*(\hat{y} \mid \boldsymbol{x}^*) = -\sum_c \bar{p}_c \frac{\log \bar{p}_c}{\log 2^c} \tag{2.5}$$

Another possible measure of uncertainty is the total variance, which is the sum of the variance of each individual class, given by:

$$V_{tot}(\hat{y} \mid \boldsymbol{x}^*) = \sum_c \frac{1}{T} \sum_{t=1}^{T} \left(\hat{p}_{c_t} - \bar{p}_c\right)^2 \tag{2.6}$$

There is no consensus in the literature on which uncertainty measurement is most appropriate when working with BNNs [42]. In their examination of the various types of uncertainty that can be measured using BNNs for computer vision tasks, however, Kendall and Gal use predictive entropy as the standard measure of prediction uncertainty for classification tasks, while predictive variance is used to estimate uncertainty in regression [33]. In this thesis, therefore, entropy (specifically normalized entropy, H_p^*) is used to make predictive uncertainty measurements for further analysis.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 3:
Methodology

This chapter is adapted from [1], previously published by the Journal of Oceanic Engineering, ©2021 IEEE

3.1 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. It is an expansion of the dataset used in a previous NPS thesis by Andrew Pfau [12]. The High-frequency Acoustic Recording Package (HARP) sensor was deployed in water 734 meters deep, with the sensor at a depth of 683 meters, 51 meters above the sea floor, and an original sample rate of 200 kilohertz [43]. Recordings were down-sampled to a 4 kilohertz sample rate for labeling and model training [12].

In parallel to the HARP deployment, automatic identification system (AIS) data was used 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. [27], in which ships are divided into four classes based upon size and one class is given to samples without ship sounds (see Table 3.1). 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.

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 3.1. Ship Classes. Source: [1]. (© 2021 IEEE)

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. 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 [12]. 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 [44]. For both datasets, a ship was deemed present if within 20 kilometers (10.7 nautical miles) of the sensor; time periods where all ships were outside 30 kilometers 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 kilometers [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 [11].

$$LCard(\mathcal{D}) = \frac{1}{N}\sum_{i=1}^{N}|Y_i| \tag{3.1}$$

Two measures of the degree of “multi-labeledness” (that is, the proportion of the dataset that is actually multi-label) are the label cardinally ($LCard$) and the label density ($LDen$).

The label cardinality of dataset $\mathcal{D}$ is the average number of labels per instance, given in Equation 3.1, while the label density is the label cardinality normalized by the number of classes, c, that make up the label space, see Equation 3.2 [45]. Calculations for the label cardinality and label density of the total HARP dataset, as well as the splits used for training, validation, and testing, are listed in 3.2. The training and validation splits are roughly the same as the overall dataset, while the test split is slightly more "multi-labeled," bolstering the generalizablilty of the multi-label performance metrics calculated for the models on the test set.

$$LDen(\mathcal{D}) = \frac{1}{c} \cdot LCard(\mathcal{D}) \tag{3.2}$$

To produce intermediate signal representations used for training, this work uses well-established, low-level acoustic signal representations in the form of mel-log spectrograms. Mel-log spectrograms are dominant features in deep learning [46] and are associated with linear-frequency spectrograms, that is, the magnitude of a Short Time Fourier Transform (STFT). They are produced by applying a mel-filterbank over STFT magnitude which effectively encapsulates frequency content at a lower dimensionality [47]. The mel-filterbank emphasizes lower frequency characteristics, which were proven to be important in underwater soundscape classification, and reduces the importance of higher frequency content which, in general, does not require high fidelity representation [12], [30].

$\mathcal{D}$	$LCard(\mathcal{D})$	$LDen(\mathcal{D})$
Total	1.437919	0.287584
Train	1.437423	0.287485
Validate	1.438696	0.287739
Test	1.441111	0.288222

Table 3.2. Label cardinality and label density for the HARP multi-label dataset. Rows represent the metrics for the overall dataset, the train split, the validation split, and the test split, respectively.

Specifically, mel-log spectrograms were computed through a STFT of 30 second labeled samples with a 500 millisecond frame size, 125 millisecond frame hop, and a Hann window function [12]. The STFT magnitude was then transformed to the mel-scale using 128 band mel-filterbank followed by log compression of the signal [47]. Labeled samples for both tasks were split further with an 80%/10%/10% ratio into training/validation/test datasets (332,760 samples/41,596 samples/41,595 samples), respectively.

Thirty seconds was chosen as the sample time for the audio clips for several reasons. First, since ships move relatively slowly (at 20 knots, a ship would move less than 34 yards, or 31 meters, in 30 seconds), their positions can essentially be treated as stationary over the course of the sample, allowing the assumption that there is little to no change in the sound emitted by the ship during the sample to hold [12]. Second, the sample time is long enough to allow multiple cycles of repeated ship noise patterns to occur, which in turn allows the model to learn these patterns. Finally, in any human-machine teaming scenario, a human operator would need some minimum baseline of information in order to detect the presence of a ship, and 30 seconds of data was chosen as reasonable approximation of this minimum [12].

In multiple studies, the examined frequency range of input signals is limited to below 200 Hertz [26]–[28]. This choice to focus on lower frequencies is supported by the findings in Arveson et al. [31] and McKenna et al. [30] that most ship noise is emitted below 1 kilohertz. While the use of mel-log spectrograms follows these findings by focusing on lower frequencies, more signal bandwidth is also utilized. By considering frequencies up to 2 kilohertz, some higher frequency sounds that can act as class discriminators, especially at shorter ranges, can be taken into account by the models while the mel-log spectrograms allow the more important lower frequencies to predominate.

3.2 Environment

The propagation, absorption and scattering of sound energy in water is highly dependent on local environmental conditions. In particular, variations in the speed of sound as it passes through the ocean can cause significant changes in transmitted and received frequencies and sound pressure levels [48]. Because of these potential changes, it is important to have a sense of how changes in the local environment might affect the performance of AI/ML systems deployed in underwater settings. The most important factors to consider when discussing local environmental conditions are pressure, temperature, and salinity [48].

These three factors combined can be used to generate a local SSP. Since ambient pressure is primarily a function of depth, sound speed in meters per second is generally expressed as a function of depth in meters, temperature in degrees Celsius, and salinity in parts per thousand [49]. Because the HARP sensor is at a static depth, pressure can generally be thought of as a constant. Salinity variations from location to location can be large in some cases (especially in places where there can be large inflows of fresh water, such as near ice shelves or river mouths). At the same location, however, ocean salinity tends to remain relatively stable [48]. Thus, the main driver in any potential temporal variations in SSP at the location of the HARP sensor are likely to be related to changes in temperature. Indeed, seasonal variations in temperature can create significant changes in SSP at the same location throughout the year [49]. Figure 3.1, taken from Kuperman and Roux [49], shows a generic shallow water seasonal SSP variation.

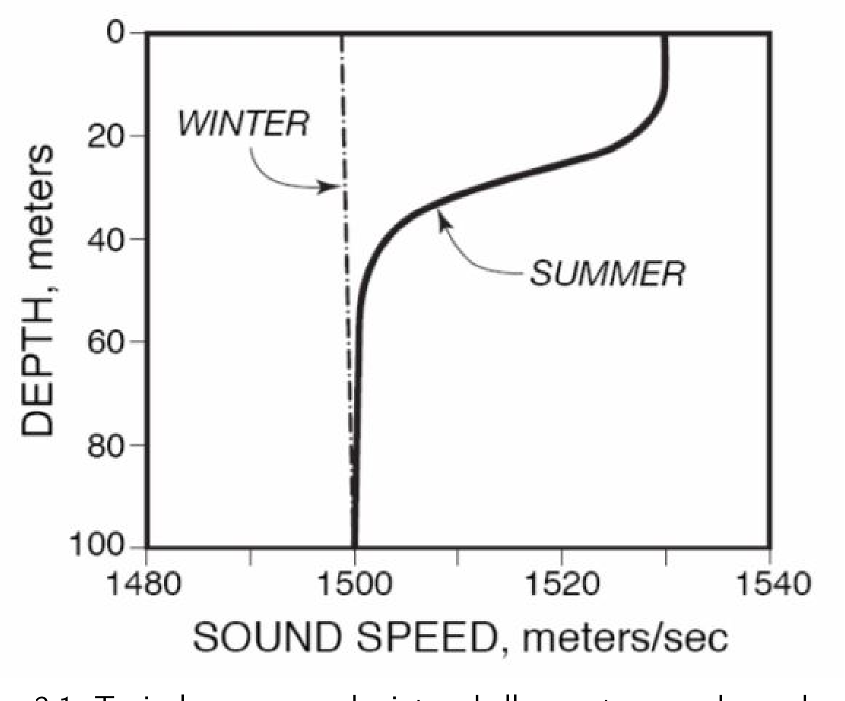
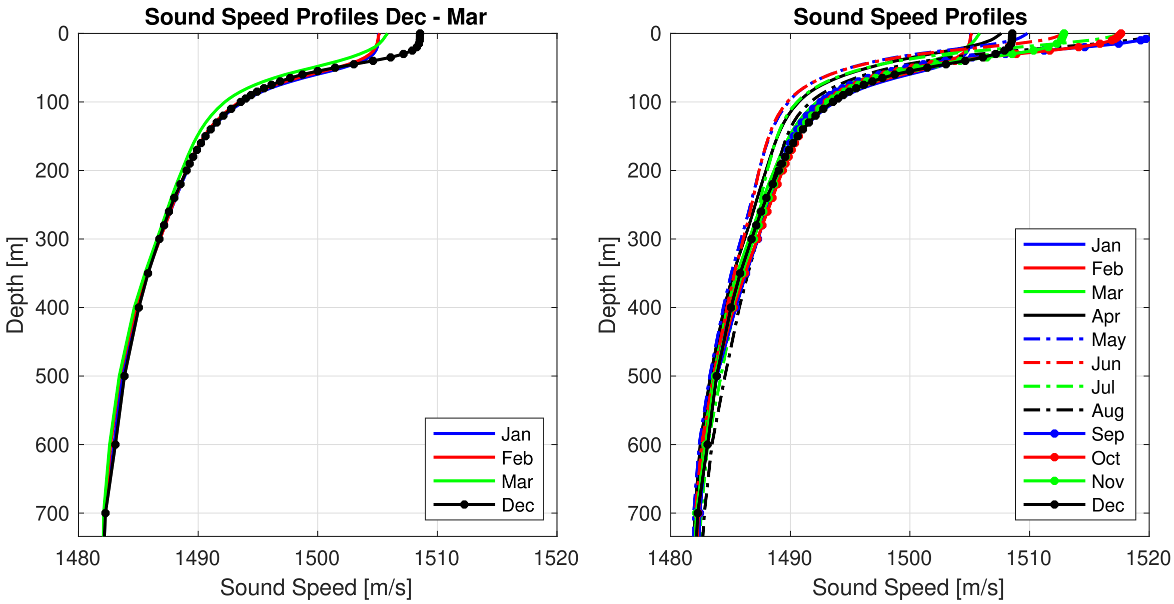


Figure 3.1. Typical summer and winter shallow water sound speed profiles. Warming causes the high speed region near the surface in the summer. Without strong heating, mixing tends to make the shallow water region isovelocity in the winter. Source: [49].

The U.S. Navy's Generalized Digital Environmental Model (GDEM) product database provides global, gridded, steady-state ocean temperature and salinity profiles [50]. Monthly temperature and salinity profiles were extracted at the nearest GDEM point (32.5N 117.75W) to the HARP location (32.666N 117.707W). These were used to derive SSPs [51] over the 12-month deployment period, see Figure 3.2.

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. Examining Figure 3.2a, from December to March low dispersion of the SSPs is observed, while for the overall time segment in Figure 3.2b dispersion between the SSPs is significantly increased.



(a) December to March

(b) December to November

Figure 3.2. Monthly SSPs at the nearest GDEM point (32.5N 117.75W) to the HARP location (32.666N 117.707W). In 3.2a, SSPs are illustrated for the December to March time frame, and in 3.2b, SSPs are illustrated for December to November. Significant dispersion can be observed in 3.2b relative to 3.2a. Source: [1]. (© 2021 IEEE)

3.3 Architecture

The popularity of CNNs is due to the state-of-the-art performance that these models achieve in large-scale image recognition tasks [23]. A desire to train deeper neural networks, potentially improving performance further, led to the development by He et al. of residual connections [16]. This paper utilizes these ResNet architectures in parallel to the custom CNN architecture to train and evaluate both deterministic and BDL models for classification tasks.

This work focuses 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 (see also Equation 2.3):

$$\hat{p}_{c_t} = softmax(\hat{f}_t(\boldsymbol{x}^*)) \tag{3.3}$$

Multi-label classification is typically constructed as multiple binary classification tasks when using a negative log likelihood loss (cross-entropy loss) [52]. A similar approach was followed by Hershey et al. [11] for audio classification using c sigmoid activation functions (σ) one over each of the c output neurons:

$$\hat{p}_{c_t} = \sigma\left(\hat{f}_t(\boldsymbol{x}^*)\right) \tag{3.4}$$

where $\hat{f}_t(\boldsymbol{x}^*) \in \mathcal{R}^c$. This is a common approach in image multi-label classification [53], [54]. 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, which in this thesis is set at 0.5 [52], [55]. The threshold was not tuned to maximize any specific metric, such as F^1 score, or to reduce the false-alarm rate. Since the ontology includes all ships and “no ship” as classes, in the case where none of the classes meet the threshold, the predicted class is selected 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 [16] architectures, standard ResNet32V1, ResNet20V1 and ResNet8V1 model configurations were tested as a deterministic baseline that was adapted to Bayesian configurations following suggestions in Tran et al. and Dillon et al. [39], [56].

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 [55]. For spectrograms derived from time-series audio data, however, typical assumptions about the invariance of image orientation do not transfer [12], thus, rectangular kernels can be applied. Multiple studies have explored the use of rectangular kernels in audio classification. Several studies use rectangular kernels in music classification [57]. Mars et al. use rectangular filters of various sizes to vary the convolution of time and frequency domains [58]. In order to adopt these ideas, rather than adjusting a ResNet architecture, a custom CNN architecture is used as shown in Figure 3.3. This architecture is directly derived from Pfau’s previous NPS thesis work [12].

Through a hyperparameter search of kernel size ratios (1:1, 2:1, 3:1, and 4:1), using a kernel size of 5 as a baseline, it was found that a 2:1 ratio is optimal and a 4:1 ratio performed the worst [12]. Hence, the final proposed kernels of 10x5 were used to apply the convolution operation on time versus frequency mel-log spectrograms. A batch normalization layer is used to normalize input mel-log spectrograms. These kernel sizes are fixed throughout every layer of the network. Based on work by Ozyildirim and Kartal [24], 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].

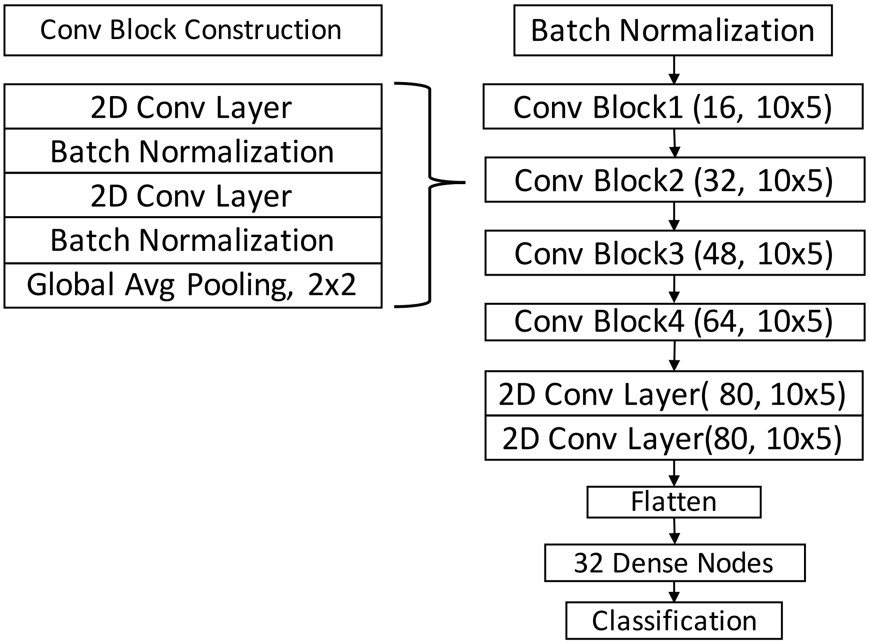


Figure 3.3. Model architecture: Each block is described by (number of filters, filter shape). Source: [12].

The architectures above were tested as described to form a deterministic baseline for both multi-class and multi-label classification tasks. They were also, however, adapted to make use of the flipout and Monte Carlo dropout BDL methods described in Section 2.3. These flipout and Monte Carlo dropout BNNs were implemented using the methodology described in an upcoming NPS thesis by Atchley [13]. In the flipout models, standard TensorFlow layers were replaced with flipout layers from the TensorFlow Probability library. For the Monte Carlo dropout models, dropout layers were added after each convolutional block (for the custom models) or each residual block (for the ResNet models) and left on during inference as well as training [13].

3.4 Evaluation 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 characteristic (ROC) curve are used to evaluate performance [55]. 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 [59]. The ability of a single test instance to be associated with multiple labels simultaneously, however, gives much greater complexity to the evaluation of multi-label performance than is present in the conventional single-label learning environment [45]. 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 [45], [60]. Label-based metrics evaluate the machine learning model for each class individually and then return the value, either micro- or macro-averaged, across all classes, see Equations 3.5 and 3.6. 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:

$$
\begin{aligned}
TP_j &= \left\{ \left| \boldsymbol{x}_i^* \mid y_j \in Y_i \wedge y_j \in f(\boldsymbol{x}_i^*), 1 \leq i \leq M \right| \right\} \\
FP_j &= \left\{ \left| \boldsymbol{x}_i^* \mid y_j \notin Y_i \wedge y_j \in f(\boldsymbol{x}_i^*), 1 \leq i \leq M \right| \right\} \\
TN_j &= \left\{ \left| \boldsymbol{x}_i^* \mid y_j \notin Y_i \wedge y_j \notin f(\boldsymbol{x}_i^*), 1 \leq i \leq M \right| \right\} \\
FN_j &= \left\{ \left| \boldsymbol{x}_i^* \mid y_j \in Y_i \wedge y_j \notin f(\boldsymbol{x}_i^*), 1 \leq i \leq M \right| \right\}
\end{aligned}
\tag{3.5}
$$

Equation 3.5 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 $\boldsymbol{x}_i^*$. Equation 3.6 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 [45].

$$
\begin{aligned}
B_{micro} &= B\left(\sum_{j=1}^{c} TP_j, \sum_{j=1}^{c} FP_j, \sum_{j=1}^{c} TN_j, \sum_{j=1}^{c} FN_j\right) \\
B_{macro} &= \frac{1}{c}\sum_{j=1}^{c} B(TP_j, FP_j, TN_j, FN_j)
\end{aligned}
\tag{3.6}
$$

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, this thesis uses subset accuracy as defined in Equation 3.7, 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 [45].

$$
Acc_{subset} = \frac{1}{M}\sum_{i=1}^{M} [[f(\boldsymbol{x}_i^*) = Y_i]] \tag{3.7}
$$

The instanced-based methods of calculating other traditional performance metrics are listed in Equation 3.8. 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 [45]. F^1 is then calculated in the usual way using the instance-based precision and recall.

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

$$Rec_{inst}(m) = \frac{1}{M}\sum_{i=1}^{M}\frac{\left|Y_i \bigcap f(\boldsymbol{x}_i^*)\right|}{|Y_i|} \tag{3.8}$$

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

The final metric discussed 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 [45]. For each test instance, HL (Equation 3.9) is the size of the symmetric difference, Δ (equivalent to "exclusive or" 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 [61]. 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 $1 - Acc$.

$$HL = \frac{1}{M}\sum_{i=1}^{M}\frac{1}{c}\left|f(\boldsymbol{x}_i^*)\Delta Y_i\right| \tag{3.9}$$

In order to give a broad picture of the comparative performance of the several models tested, for both multi-class and multi-label models this paper reports the macro-averaged and weighted-averaged label-based precision, recall, and F^1 score as calculated in Equation 3.6, as well as the macro-averaged AUC and the HL (see Equation 3.9). For multi-label performance, it also reports label-based micro-averaged and instance-based precision, recall and F^1 score as shown in Equations 3.6 and 3.8. Accuracy is reported as discussed in the examination of Equations 3.6 and 3.7 above.

3.5 Experiments

The custom CNN architecture as well as ResNet32V1, ResNet20V1 and ResNet8V1 architectures formed the bases for the models tested. Each base was used to create six separate model architectures. These model architectures were multi-class deterministic, multi-class flipout, multi-class Monte Carlo dropout, multi-label deterministic, multi-label flipout, and multi-label Monte Carlo dropout, for a total of 24 separate base-architecture configurations (four bases times six model architectures). These separate models were then trained on the full multi-class or multi-label datasets, respectively. For evaluation, deterministic models made predictions on each of the instances in the 10% of the data held out as a test set using Equation 3.3 for multi-class and Equation 3.4 (using the thresholds and modifications discussed in Section 3.3) for multi-label. BNN models made 50 inferences on each test instance, with the results averaged across classes before calculating uncertainty and applying the appropriate activation function.

All of the model architectures evaluated were developed using the same training strategy. This standardization ensures the fairness of benchmarking. The Adam optimization algorithm was used for training with 0.001 as the initial learning rate. This starting learning rate was discovered to be the most effective after running experiments on both custom CNN and ResNet models using starting learning rates from 1 to 0.0001. Learning rate annealing [62] is employed by monitoring validation accuracy and reducing the learning rate by a factor of 10 if the validation accuracy was not increasing for 50 consecutive epochs. To regularize for overfitting, in addition to the methods mentioned in Section 3.3, an early stopping strategy was utilized [55]. Overall, training was terminated at a maximum of 500 epochs. NVIDIA RTX 8000 48GB GPU graphics cards were used for distributed model training and model inference.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 4:
Results

This chapter is adapted from [1], previously published by the Journal of Oceanic Engineering, ©2021 IEEE

4.1 Overall Results

The experiments described above examined the performance of traditional deterministic (non-Bayesian), Monte Carlo dropout, and flipout versions of both the 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 Monte Carlo 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. Models of each of the versions were trained on the full HARP dataset for multi-class and multi-label classification, respectively.

	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 4.1. Multi-class performance metrics. Source: [1]. (© 2021 IEEE)

The results for multi-class classification are reported in Table 4.1 while the results for multi-label classification are in Table 4.2. 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 Monte Carlo dropout BNN, generally reflective of the performance enhancements seen in ensemble learning (Monte Carlo dropout can be viewed as an ensemble classifier where each inference is a different model) [15]. The exception to this was in multi-label classification, where the deterministic and Monte Carlo dropout versions performed almost identically, varying at most by 2% in any one metric. Based on uncertainty measurements discussed below, a speculation can be made 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 Monte Carlo dropout model was unable to make more accurate predictions than the deterministic model.

	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 4.2. Multi-label performance metrics. Source: [1]. (© 2021 IEEE)

4.2 Uncertainty Filtering

Much research in AI/ML simply evaluates model predictions on a test dataset and computes some sort of measure of the amount of correct predictions versus incorrect predictions in a straightforward manner [40] (the metrics of Section 3.4 all attempt to do exactly this). While these methods are certainly important to measuring and improving model performance, they do not perfectly capture the full range of options available to a classification or diagnostic system. In many tasks for which AI/ML systems, such as traditional CNNs, are used, like medical diagnoses or language processing, humans doing the same task would have options available to them besides simply making a prediction [40]. These other options include asking for more information, like longer samples or additional images, or asking for a second opinion, especially of someone with significant expertise in the relevant field. Figure 4.1 illustrates a key benefit of the use of BNNs 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 the model, but it can also be used to triage only the most ambiguous classifications for further analysis by experts [40].

Examining the multi-class Monte Carlo dropout model in Figure 4.1 reveals that filtering out all samples with H_p^* greater than 0.375 retains about 80% of the samples but improves the weighted F^1 score from 0.82 to 0.90. For the multi-label Monte Carlo 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. Figure 4.2 demonstrates the reason for the performance boost gained by filtering. There is a much larger number of predictions for which the model is relatively certain, and these are predictions which the model is also more likely to get correct. Filtering out the more uncertain predictions raises performance scores while not affecting many of the predictions overall.

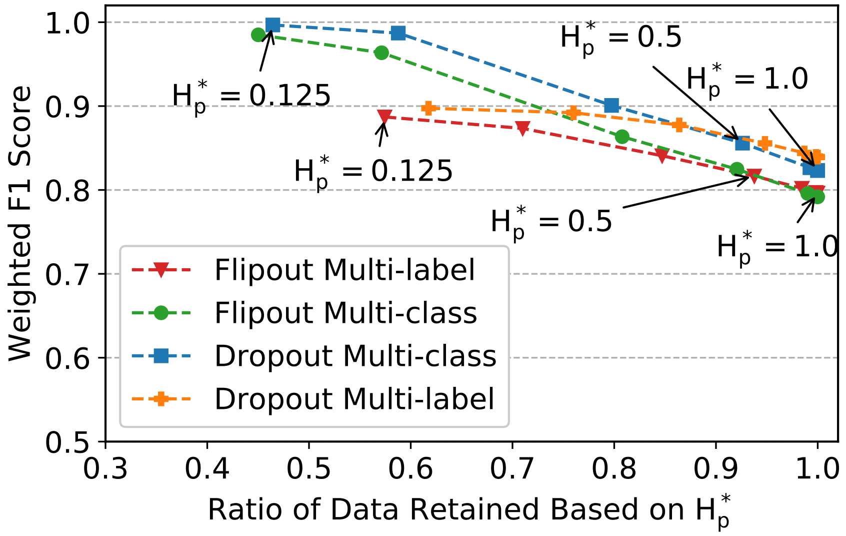


Figure 4.1. Each point represents the model's weighted average F^1 score vs. the ratio of overall number of samples retained when filtering out all samples above a certain predictive uncertainty value (measured by H_p^*). Source: [1]. (© 2021 IEEE)

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.

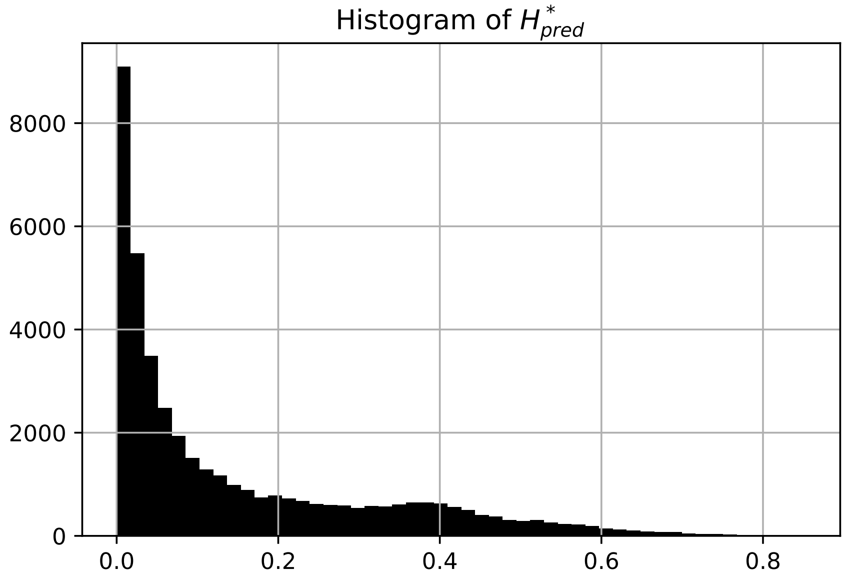


Figure 4.2. Histogram of H^*_p for each prediction made on the test dataset by the custom architecture multi-label Monte Carlo dropout BNN model. The X-axis is the value of H^*_p and the Y-axis is the number of instances.

Both Monte Carlo dropout and flipout Bayesian architectures produce calibrated uncertainties, see Figure 4.1. However, the overall results indicate better performance for Monte Carlo dropout model architectures across the evaluated metrics for both multi-class and multi-label classification tasks as shown in Table 4.1 and Table 4.2. Given that the Monte Carlo dropout models have significantly fewer parameters and take less computational time than their flipout counterparts, Monte Carlo dropout is a promising option for sonar classification, especially for use on unmanned vehicles and other embedded systems.

4.3 Case Study I

To explore how all of these results fit together in practice, Case Study I was developed. In this case study, a ship was chosen and the classification predictions made by one of the models as that ship transited past the HARP sensor were examined in detail. The model used was the custom multi-class Monte Carlo dropout BNN trained on the full HARP dataset. The ship selected was a car carrier which sailed near the HARP sensor over about a four-hour period on February 25, 2013. The corresponding AIS information was used to build a range versus time plot and examined the model's predictive output and uncertainty along the track as shown in Figure 4.3. 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, the same effect is seen, with less accurate and less certain predictions farther from the sensor. Figure 4.4 shows the same information as Figure 4.3 overlaid on a map with the target's latitude and longitude plotted for each prediction.

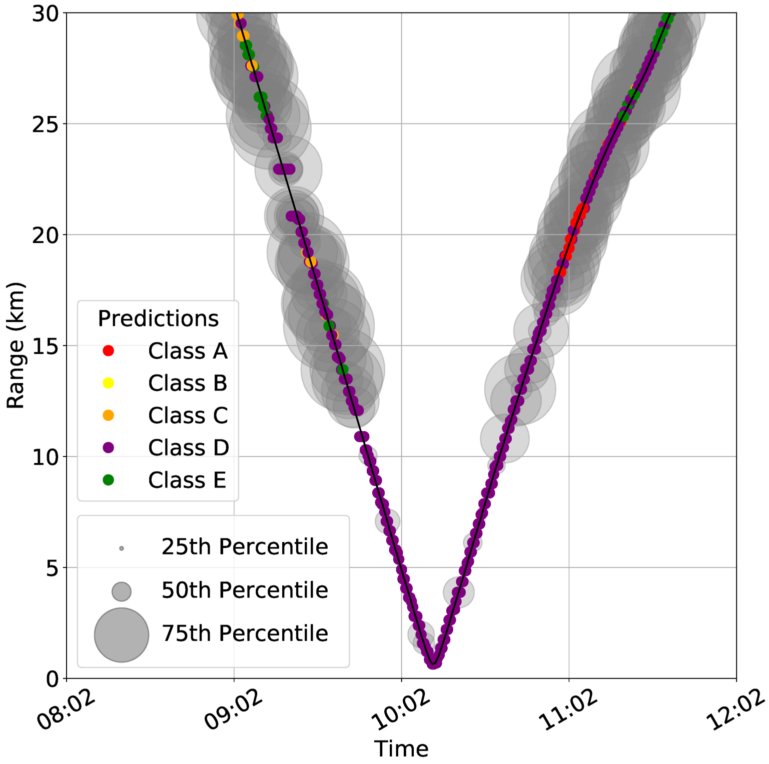


Figure 4.3. Case Study I - February 25 2013, Car Carrier; range vs. time plot of a known target AIS track overlaid with BNN classification output and predictive entropy, H_p. Classification outputs are color coded relative to the class where Car Carrier (classD) is correctly classified with magenta. 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 Monte Carlo dropout BNN trained on the full HARP dataset (see third column, Custom/Drop, in Table 4.1). Source: [1]. (© 2021 IEEE)

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 [63]. 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 the first missed predictions are observed. 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 in Case Study I, as well as better performance at closer ranges, matches what would be expected in real-world conditions and demonstrates the model's ability to reflect reality in its performance and uncertainty measurements.

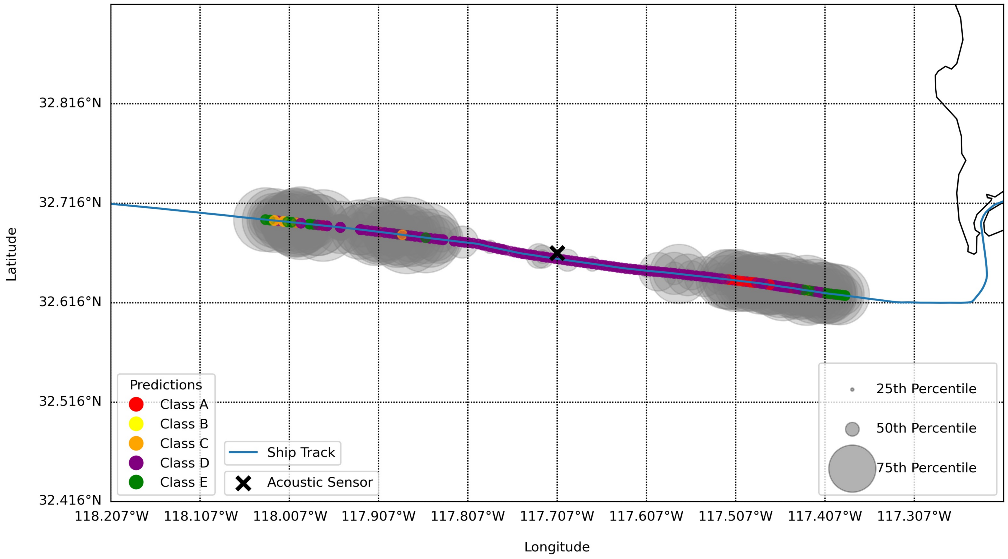


Figure 4.4. Case Study I - February 25 2013, Car Carrier; geographic AIS location plot of a known target AIS track overlaid with BNN classification output and predictive entropy, H_p. Classification outputs and entropy calculations are the same as used in Figure 4.3.

4.4 Case Study II

As another case study, additional versions of the custom model were trained on a seasonal subset of the data. Case Study II looked at the performance of the general multi-class models trained on the whole year's data (see Table 4.1) 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 4.3. 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.

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 4.3. 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 the 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. Source: [1]. (© 2021 IEEE)

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 Figure 3.2, seasonal changes to the water column's SSP affect 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.

CHAPTER 5: Conclusion

This chapter is adapted from [1], previously published by the Journal of Oceanic Engineering, ©2021 IEEE

The classification of underwater sounds is of great interest to several communities 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. The 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 developing field, which can have important implications on the proliferation of deep learning models in production.

The presented results and associated analysis are 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, the second case study 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. Accurate, automated sonar classification enables more autonomy in unmanned systems, while increasing efficiency and reducing cognitive load on human sonar operators. The addition of uncertainty measurements further enhances the effectiveness of human-computer teaming in this domain. Overall, the proposed approaches can have a significant impact on the autonomous monitoring of ocean resources through passive sonar, as well as on the operational effectiveness of multiple air, surface, and subsurface U.S. Navy platforms.

5.1 Future Work

There are several research areas in which this work could be extended. The first of these is in model architecture and its capabilities. Further study of the construction of the custom CNN architecture on which the BNN models are based could yield better performance. While the hyperparameter spaces for both the CNN architecture and the BNNs architectures developed on top of it were explored, this search was by no means exhaustive. More hyperparameter tuning is likely a fruitful avenue of further investigations. Another capability that should be added in future versions is the ability to detect multiple instances of the same class of ship simultaneously. Additionally, while classifications are useful, in order to truly be operational, the system must also be capable of conducting reasonably accurate bearing and range regressions, which would require extensive architecture modifications.

The next major area for continued inquiry is in the data used for training and testing the models. Data augmentation techniques have yielded impressive performance improvements in many areas of AI/ML, but especially in computer vision [64]. Since the models in this thesis use images (the mel-log spectrograms generated from the raw audio data) as their inputs, this approach is a particularly promising one. Furthermore, the HARP sensor used here provided only one channel of audio data. The use of a multi-channel sensor could provide additional opportunities to apply methods from other areas of AI/ML research.

Finally, while the size and scope of the dataset used here represents a significant improvement from most previous work, it is still representative of only one locale and environment in a large and diverse global ocean. In addition to investigating temporal variations and "seasonal expert" classifiers, as mentioned in Section 4.4, gathering data from other locations, especially ones with substantially different environmental conditions and SSPs, for use with the models presented in this work, as well as other models developed independently, will be critical to the development of the generalizable models necessary for real-world deployments.

List of References

[1] B. Beckler, A. Pfau, M. Orescanin, S. Atchley, J. E. Joseph, C. W. Miller, and T. Margolina, "Multi-label classification of heterogeneous underwater soundscapes with Bayesian deep learning," *IEEE Journal of Oceanic Engineering*, 2021.

[2] S. Lohr, "IBM is counting on its bet on Watson, and paying big money for it," *The New York Times*, Oct. 2016. Available: https://www.nytimes.com/2016/10/17/technology/ibm-is-counting-on-its-bet-on-watson-and-paying-big-money-for-it.html?emc=edit_th_20161017&nl=todaysheadlines&nlid=62816440

[3] A. L'Heureux, K. Grolinger, H. F. Elyamany, and M. A. M. Capretz, "Machine learning with big data: Challenges and approaches," *IEEE Access*, vol. 5, pp. 7776–7797, 2017.

[4] Joint Artificial Intelligence Center, "About the JAIC - JAIC," Aug. 2020. Available: https://www.ai.mil/about.html

[5] 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, June 2004.

[6] 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. Available: http://ieeexplore.ieee.org/document/8121894/

[7] 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, July 2019. Available: http://arxiv.org/abs/1907.13188

[8] 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.

[9] 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. Available: http://ieeexplore.ieee.org/document/7761112/

[10] 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. Available: https://www.mdpi.com/2079-9292/9/11/1972

[11] 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. Available: http://arxiv.org/abs/1609.09430

[12] A. M. Pfau, “Multi-label classification of underwater soundscapes using deep convolutional neural networks,” M.S. thesis, Dept. of Comp. Sci., NPS, Monterey, CA, 2020. Available: https://calhoun.nps.edu/handle/10945/66705

[13] S. Atchley, “Passive sonar classification using active Bayesian deep learning with a vector sensor,” unpublished.

[14] Y. Gal and Z. Ghahramani, “Dropout as a Bayesian approximation: Representing model uncertainty in deep learning,” in *Proceedings of The 33rd International Conference on Machine Learning*, Oct. 2016, pp. 1050–1059. Available: http://arxiv.org/abs/1506.02142

[15] H. Wang and D.-Y. Yeung, “A survey on Bayesian deep learning,” *ACM Computing Surveys*, vol. 53, no. 5, pp. 1–37, Oct. 2020. Available: https://dl.acm.org/doi/10.1145/3409383

[16] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE, June 2016, pp. 770–778. Available: http://ieeexplore.ieee.org/document/7780459/

[17] K. He, X. Zhang, S. Ren, and J. Sun, “Identity mappings in deep residual networks,” in *Computer Vision – ECCV 2016*, vol. 9908, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 630–645, series title: *Lecture Notes in Computer Science*. Available: http://link.springer.com/10.1007/978-3-319-46493-0_38

[18] A. Ramesh, C. Kambhampati, J. Monson, and P. Drew, “Artificial intelligence in medicine,” *Annals of The Royal College of Surgeons of England*, vol. 86, no. 5, pp. 334–338, Sep. 2004. Available: http://www.ingentaselect.com/rpsv/cgi-bin/cgi?ini=xref&body=linker&reqdoi=10.1308/147870804290

[19] A. L. Samuel, “Some studies in machine learning using the game of checkers,” *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210–229, July 1959. Available: http://ieeexplore.ieee.org/document/5392560/

[20] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 2nd ed. Beijing; Sebastopol, CA: O’Reilly Media, Inc, 2019.

[21] S. C. Shapiro, *Artificial Intelligence (AI)*. GBR: John Wiley and Sons Ltd., 2003, p. 89–93.

[22] 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. Available: http://asa.scitation.org/doi/10.1121/1.5133944

[23] 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, publisher: AcM New York, NY, USA.

[24] 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. Available: https://www.seekdl.org/conferences/paper/details/8220

[25] L. Nanni, Y. M. G. Costa, R. L. Aguiar, R. B. Mangolin, S. Brahnam, and C. N. Silla, "Ensemble of convolutional neural networks to improve animal audio classification," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2020, no. 1, p. 8, Dec. 2020. Available: https://asmp-eurasipjournals.springeropen.com/articles/10.1186/s13636-020-00175-3

[26] 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. Available: http://www.intechopen.com/books/self-organizing-maps-applications-and-novel-algorithm-design/ship-s-hydroacoustics-signatures-classification-using-neural-networks

[27] 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. Available: https://linkinghub.elsevier.com/retrieve/pii/S0003682X16301566

[28] 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. Available: http://asa.scitation.org/doi/10.1121/1.5010064

[29] 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. Available: https://asa.scitation.org/doi/10.1121/10.0003361

[30] 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. Available: http://asa.scitation.org/doi/10.1121/1.3664100

[31] 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. Available: http://asa.scitation.org/doi/10.1121/1.428344

[32] W. M. Bolstad and J. M. Curran, *Introduction to Bayesian Statistics*, 3rd ed. Hoboken, N.J.: John Wiley & Sons, 2017. Available: http://ndl.ethernet.edu.et/bitstream/123456789/37609/1/William%20M.%20Bolstad_2017.pdf

[33] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" in *Advances in Neural Information Processing Systems*, Long Beach, CA, Oct. 2017. Available: http://arxiv.org/abs/1703.04977

[34] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American Statistical Association*, vol. 112, no. 518, pp. 859–877, Apr. 2017. Available: http://arxiv.org/abs/1601.00670

[35] M. Orescanin, V. Petkovic, S. W. Powell, B. R. Marsh, and S. C. Heslin, "Bayesian deep learning for passive microwave precipitation type detection," *IEEE Geoscience and Remote Sensing Letters*, pp. 1–5, 2021. Available: https://ieeexplore.ieee.org/document/9474574/

[36] Y. Wen, P. Vicol, J. Ba, D. Tran, and R. Grosse, "Flipout: Efficient pseudo-independent weight perturbations on mini-batches," in *Conference Track Proceedings*, Vancouver, Apr. 2018. Available: http://arxiv.org/abs/1803.04386

[37] 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*, pp. 1–8, 2021.

[38] 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. Curran Associates, Inc., 2015, vol. 28. Available: https://proceedings.neurips.cc/paper/2015/file/bc7316929fe1545bf0b98d114ee3ecb8-Paper.pdf

[39] 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*, Nov. 2017. Available: http://arxiv.org/abs/1711.10604

[40] A. Filos, S. Farquhar, A. N. Gomez, T. G. J. 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," in *4th workshop on Bayesian Deep Learning*, Vancouver, Dec. 2019. Available: http://arxiv.org/abs/1912.10481

[41] 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.

[42] O. Dürr, B. Sick, and E. Murina, *Probabilistic Deep Learning: with Python, Keras and TensorFlow Probability* (Exercises in Jupyter Notebooks). Shelter Island, NY: Manning Publications Co, 2020.

[43] 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. Available: http://ieeexplore.ieee.org/document/4231090/

[44] VesselFinder, "Vessel Database," June 2021. Available: https://www.vesselfinder.com/

[45] 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. Available: http://ieeexplore.ieee.org/document/6471714/

[46] 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.

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

[48] X. Lurton, *An Introduction to Underwater Acoustics: Principles and Applications*, 2nd ed. (Springer-Praxis books in geophysical sciences). Berlin: Springer, 2010.

[49] W. Kuperman and P. Roux, "Underwater acoustics," in *Springer Handbook of Acoustics*, T. Rossing, Ed. New York, NY: Springer New York, 2007, pp. 149–204. Available: http://link.springer.com/10.1007/978-0-387-30425-0_5

[50] R. Allen Jr, "Naval Oceanographic Office Generalized Digital Environmental Model (GDEM) product database. Object name NODC Accession 9600094." Available: https://www.ncei.noaa.gov/archive/accession/download/9600094

[51] 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.

[52] H. Guo, K. Zheng, X. Fan, H. Yu, and S. Wang, "Visual attention consistency under image transforms for multi-label image classification," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA: IEEE, June 2019, pp. 729–739. Available: https://ieeexplore.ieee.org/document/8954355/

[53] T. Durand, N. Mehrasa, and G. Mori, "Learning a deep ConvNet for multi-label classification with partial labels," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA: IEEE, June 2019, pp. 647–657. Available: https://ieeexplore.ieee.org/document/8954216/

[54] J. Wehrmann, R. Cerri, and R. Barros, "Hierarchical multi-label classification networks," in *International Conference on Machine Learning*. PMLR, July 2018, pp. 5075–5084, iSSN: 2640-3498. Available: http://proceedings.mlr.press/v80/wehrmann18a.html

[55] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (Adaptive Computation and Machine Learning). Cambridge, MA: The MIT Press, 2016.

[56] D. Tran, M. W. Dusenberry, M. van der Wilk, and D. Hafner, "Bayesian layers: A module for neural network uncertainty," in *Advances in Neural Information Processing Systems*, Vancouver, Dec. 2019, pp. 14 660–14 672. Available: https://papers.nips.cc/paper/2019/file/154ff8944e6eac05d0675c95b5b8889d-Paper.pdf

[57] 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. Available: http://ieeexplore.ieee.org/document/7952601/

[58] 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. Available: http://hdl.handle.net/2451/60748

[59] M. Grandini, E. Bagli, and G. Visani, "Metrics for multi-class classification: An overview," *arXiv*, Aug. 2020. Available: https://arxiv.org/abs/2008.05756

[60] S. Godbole and S. Sarawagi, “Discriminative methods for multi-labeled classification,” in *Advances in Knowledge Discovery and Data Mining*, vol. 3056, 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, pp. 22–30, series Title: Lecture Notes in Computer Science. Available: http://link.springer.com/10.1007/978-3-540-24775-3_5

[61] G. Tsoumakas and I. Katakis, “Multi-label classification: An overview,” *International Journal of Data Warehousing and Mining*, vol. 3, no. 3, pp. 1–13, July 2007. Available: http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/jdwm.2007070101

[62] 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*, Vancouver, Dec. 2019, pp. 11 674–11 685. Available: https://papers.nips.cc/paper/2019/file/bce9abf229ffd7e570818476ee5d7dde-Paper.pdf

[63] 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. Available: http://asa.scitation.org/doi/10.1121/1.4739251

[64] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *Journal of Big Data*, vol. 6, no. 1, p. 60, Dec. 2019. Available: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0197-0

THIS PAGE INTENTIONALLY LEFT BLANK

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California