NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

AUTOMATING VESSEL DETECTION WITH PASSIVE SONAR SIGNALS AND CONVOLUTIONAL NEURAL NETWORKS

by

John W. Kim

September 2020

Thesis Advisor:	Robert L. Bassett
Second Reader:	Lyn R. Whitaker

THIS PAGE INTENTIONALLY LEFT BLANK

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

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2020	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE AUTOMATING VESSEL DETECTION WITH PASSIVE SONAR SIGNALS AND CONVOLUTIONAL NEURAL NETWORKS	**5. FUNDING NUMBERS**
6. AUTHOR(S) John W. Kim	
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)

In recent years, new acoustic stealth platforms, which have the potential to operate invisibly from human sonar operators, have emerged from near-peer competitor nations. In response to the challenges presented by acoustic detection and classification in adversarial marine environments, we proposed a novel application of convolutional neural networks for autonomous passive sonar analysis. Neural networks have made significant strides in multiple fields due to their powerful image recognition abilities. Using time and location information from Automatic Identification System (AIS) data provided by the U.S. Coast Guard, we labeled acoustic signal data recorded by an underwater hydrophone to nearby vessels. We then converted the labeled acoustic data into spectrogram images detailing the frequency, amplitude, and timesteps. With these spectrogram images, we attempted to train several convolutional neural networks to recognize images indicating the presence of maritime vessels. Our results exhibited severe overtraining and unreliable classification of the spectrogram images. We then explored the possibility of converting the spectrogram images to mean frequency vectors and applying other machine-learning algorithms to these vectors. These algorithms produced much more promising classification rates than those of the convolutional neural networks. We hope that our research may be further developed in the future for practical applications in autonomous acoustic classification.

14. SUBJECT TERMS Monterey Bay Aquarium Research Institute, MBARI, sonar, passive sonar, neural network, convolutional neural network, CNN, artificial intelligence, spectrogram, deep learning, machine learning, Automatic Identification System, AIS, image recognition, AlexNet, ShuffleNet, Automatic Identification System, AIS, Gaussian Process, K-nearest neighbor, KNN, Monterey Accelerated Research System, MARS, mean frequency vector, Quadratic Discriminant Analysis, QDA, support vector machine, SVM, MLP	**15. NUMBER OF PAGES** 63
	16. PRICE CODE

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

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

AUTOMATING VESSEL DETECTION WITH PASSIVE SONAR SIGNALS AND CONVOLUTIONAL NEURAL NETWORKS

John W. Kim
Lieutenant, United States Navy
BS, Virginia Tech, 2013

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September 2020

Approved by: Robert L. Bassett
Advisor

Lyn R. Whitaker
Second Reader

W. Matthew Carlyle
Chair, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

In recent years, new acoustic stealth platforms, which have the potential to operate invisibly from human sonar operators, have emerged from near-peer competitor nations. In response to the challenges presented by acoustic detection and classification in adversarial marine environments, we proposed a novel application of convolutional neural networks for autonomous passive sonar analysis. Neural networks have made significant strides in multiple fields due to their powerful image recognition abilities. Using time and location information from Automatic Identification System (AIS) data provided by the U.S. Coast Guard, we labeled acoustic signal data recorded by an underwater hydrophone to nearby vessels. We then converted the labeled acoustic data into spectrogram images detailing the frequency, amplitude, and timesteps. With these spectrogram images, we attempted to train several convolutional neural networks to recognize images indicating the presence of maritime vessels. Our results exhibited severe overtraining and unreliable classification of the spectrogram images. We then explored the possibility of converting the spectrogram images to mean frequency vectors and applying other machine-learning algorithms to these vectors. These algorithms produced much more promising classification rates than those of the convolutional neural networks. We hope that our research may be further developed in the future for practical applications in autonomous acoustic classification.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

AIS Automatic Identification System

COI Contact of Interest

CNN Convolutional Neural Network

DoD Department of Defense

DFT Discrete Fourier Transformation

FC Fully Connected

FC-NN Fully Connected Neural Network

GP Gaussian Process

KNN K-Nearest Neighbor

MARS Monterey Accelerated Research System

MBARI Monterey Bay Aquarium Research Institute

MFV Mean Frequency Vector

MLP Multilayer Perceptron

NN Neural Network

NPS Naval Postgraduate School

QDA Quadratic Discriminant Analysis

SVM Support Vector Machine

USCG United States Coast Guard

USN United States Navy

USW Undersea Warfare

THIS PAGE INTENTIONALLY LEFT BLANK

Executive Summary

For U.S. Navy commanders, a keen perception of the maritime battlespace is often a crucial component for operational success. One of the most extensively used methods to attain awareness of the operational environment is acoustic sensing through passive sonar systems. In recent years, highly advanced acoustic stealth platforms have emerged from near-peer competitor nations which have the potential to operate invisibly from human sonar operators. In response to the challenges presented by acoustic detection and classification in adversarial marine environments, we propose a novel approach of convolutional neural networks for autonomous passive sonar analysis. Neural networks have made significant strides in multiple fields due to their powerful image recognition abilities, and we explored their applicability to vessel detection through spectrogram feature learning.

With the assistance of the Monterey Bay Aquarium Research Institute (MBARI), the Naval Postgraduate School's (NPS) Department of Physics, and the U.S. Coast Guard Navigation Center, we created a dataset comprised of acoustic spectrogram images which indicate whether one or more vessels were within the vicinity of an underwater hydrophone during a specified timeframe. This hydrophone, operated by MBARI and the NPS Physics Department, provided several weeks of continuous acoustic data from an offshore facility several miles off the coast of Monterey. We associated the hydrophone's location and acoustic timestamp data with vessel information provided by the U.S. Coast Guard's collection of Automatic Identification System (AIS) data. This allowed us to create a dataset of several thousand spectrogram images, each containing a 5-minute time window depicting the frequency and amplitude of the local acoustic environment. Based on the associated AIS data, these spectrograms were labeled according to whether they indicated the presence of acoustic signal information for nearby vessels.

The primary focus of our endeavors was based on two pre-existing convolutional neural network architectures, AlexNet and ShuffleNet. AlexNet was selected due to its prestigious reputation in the field of artificial image recognition, and we believed that its emphasis on depth of layers for feature extraction would be beneficial to spectrogram classification. We chose to use ShuffleNet due to its computational efficiency and ability to maintain a high level of accuracy for larger scaled datasets. However, once we began the feature learning

process, neither of these architectures produced favorable results. AlexNet and ShuffleNet both tended to overtrain and were unable to accurately classify the spectrograms with any reasonable degree of accuracy.

We then decided to conduct further experiments with other machine learning algorithms: Quadratic Discriminant Analysis, decision trees, Gaussian Process classifiers, K-nearest neighbors, support vector machines, multilayer perceptron neural networks, and logistic regression. Instead of using the original spectrogram images, we based the training of these new algorithms with vectorized versions of the spectrograms by taking the mean frequency across the time axis of each spectrogram image. Through these mean frequency vectors, our selected machine learning algorithms produced much more promising results. In particular, the support vector machine and K-nearest neighbor algorithms generated impressively high classification accuracy rates, while logistic regression exhibited the least evidence of overtraining.

Our experiments with convolutional neural networks and other machine learning algorithms garnered numerous insights into the potential of automated acoustic detection. We hope that our research may be further developed in the future for practical applications in automating the detection and classification process of acoustic signals.

Acknowledgments

My time at the Naval Postgraduate School has been an incredible journey. I would like to express my sincerest thanks to the mentors and peers who have constantly supported me and made my accomplishments possible.

To my advisor, Dr. Robert Bassett, I offer my deepest gratitude for your guidance and endless patience. You are truly a master of your craft, and this project would not have been possible without you. I cannot express enough how lucky I feel to have received your mentorship.

To my second reader, Dr. Lyn Whitaker, thank you for your advice and assistance throughout this process. Our sessions together gave me the tools and insights I needed to understand machine learning on a deeper, more profound level.

To my cohort, thank you for your friendship and support through the rigors of the Operations Research curriculum. There were too many nights when I felt overwhelmed by our course load, only for one of you to raise my spirits through heroic last-minute assistance or even just giving a moment of laughter and friendship. The bonds we forged were truly special, and I look forward to seeing all of you in the fleet.

And lastly, to Jenn, who provided me with endless love and support. Thank you for reminding me of what's really important in life, and for inspiring me to be a better person.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction

1.1 Problem Statement

In the world of undersea warfare (USW), maritime nations are continuously striving to make their naval platforms as undetectable as possible. Navies invest heavily in diminishing the visibility of their ships, submarines, and mines while also developing tools and techniques to find and track adversary units. The most common method of detection is passive sonar, in which a sensor system "listens" for underwater acoustic signals which is then interpreted by a human operator who determines whether a signal should be associated with a contact of interest (COI). Sonar detection and classification has been practiced by the U.S. Navy for decades, but new technologies are making it more and more difficult for our operators to detect adversary platforms. As our own systems become more advanced, machine learning is quickly emerging as an essential tool in assisting human sonar operators find COIs, and may even be used for autonomous detection via unmanned platforms. In the past several years, convolutional neural networks (CNN) have shown great success in image recognition software. We theorized that if acoustic data were converted to a series of spectrogram images, a CNN may be able to ascertain distinguishable features which would allow it to learn to recognize when a spectrogram indicated the presence of a vessel. With this possibility in mind, we set out to explore the feasibility of applying these networks to real-world data to explore the potential of acoustic detection through machine learning.

1.2 Relevance

The operation of autonomous agents requires software and hardware which facilitate understanding and responding to complex external environments. To cope with this complexity, researchers and engineers rely on increasingly sophisticated methods for translating sensor data into automated decisions for interaction within the environment, otherwise known as signal detection and classification. In Department of Defense (DoD) applications, autonomous agents often operate in an environment influenced by an adversary, which presents additional challenges. In recent years, several new stealth platforms and technologies have

emerged which have the potential to threaten U.S. assets. Examples include Russia's newest nuclear guided-missile submarine, the *Severodvinsk* (Majumdar 2014); the People's Republic of China's new air-independent *Yuan* diesel-electric submarine (Majumdar 2018); and Iran's considerable and proven-deadly mine inventory (LaGrone 2019).

In addition to the system's technical capabilities, naval units must also rely on a sonar operator's personal level of skill. In the Mackworth Clock Experiment, Norman Mackworth found that a human operator can focus on and accurately respond to a given task for approximately 30 minutes before his or her concentration skills begin to degrade (Lichstein et al. 2000). For a sonar operator scanning the ocean for hours at a time, a COI could easily go undetected without constant awareness. An operator's skill level can be quantified through a parameter known as a recognition differential. The difference between the recognition differential and the acoustic characters tics of both the unit and the COI produces the estimated likelihood that the COI will be detected, with a higher recognition differential will result in a higher likelihood of detection (Wagner and Mylander 1999) With the ever-evolving threats the U.S. Navy faces, raising an operator's recognition differential can play a key part in detecting COIs. Automating the acoustic tracking process may prove to be a viable method to increase recognition differential and enhance detection capability.

1.3 Literature Review

Several previous published studies serve as foundational references for our research with convolutional neural networks.

Moore (1991) conducted early work on the plausibility of using neural networks for sonar classification. His thesis describes his efforts to build a basic back-propagating neural network to a simulated dataset and concluded that future practical applications were possible given careful selection of network parameters.

Preliminary studies on convolutional neural networks for spectrogram classification was later conducted by Rippel et al. (2015), who focused on the use of spectral parameterization and spectral pooling to reduce dimensionality during training and speed up processing time. Rippel et al. (2015)'s study also demonstrated the wealth of possibilities convolutional neural networks can provide when used in conjunction with Fourier transforms embedded to a frequency domain.

Further research was conducted by Becker et al. (2018), which exhibited the effectiveness of several convolutional neural networks when applied to a data set consisting of audio recordings of individuals speaking numerical digits. Their data was collected in a carefully controlled manner and labeled without ambiguity. Additionally, they applied their convolutional neural networks to both converted spectrogram images and raw audio waveforms, whereas our study focused on spectrogram classification.

Salamon and Bello (2017) explored training CNNs on urban soundscapes, focusing on specific sound classes such as street music, car horns, and barking dogs, among others. Salamon and Bello (2017)'s study included a secondary objective of using injected augmentation to overcome data scarcity. They concluded that inputting data augmentation was a plausible means to train a CNN.

In 2017, Niu et al. (2017) published their own research on the application of machine learning algorithms on ship detection and localization. Their study utilized a feed-forward, a support vector machine, and random forest methods on three pre-selected ships transiting the Santa Barbara Channel. Using an acoustic array submerged at a depth of 600 meters, they uncovered promising results for varying speeds of their vessels at distances of up to 10 kilometers.

1.4 Design and Execution

In response to the challenges presented by acoustic detection in adversarial marine environments, we propose a novel method of acoustic detection with the use of a convolutional neural network. Neural networks have made significant strides in multiple fields due to their powerful image recognition abilities. We associated a dataset comprised of acoustic data from a stationary hydrophone operated by the Monterey Bay Aquarium Research Institute (MBARI), with Automatic Identification System (AIS) data provided by the U.S. Coast Guard (USCG). This combined dataset allowed us to label acoustic signals recorded by the MBARI hydrophone as containing vessel acoustic information. We then converted the labeled acoustic data into spectrogram images detailing the signal's frequency, amplitude, and timesteps. With these spectrogram images, a convolutional neural network may be developed and trained to recognize maritime vessels. The goal of our research was to demonstrate that a CNN can be trained to distinguish acoustic signals in a marine envi-

ronment. To that end, we trained CNN architectures with the labeled acoustic spectrogram images to recognize whether the image indicated seagoing vessels operating nearby. The trained CNNs were then used to classify a test dataset to validate the effectiveness of the training. Our results delivered a quantitative measure of the CNN architecture's capability by identifying the test set's number of correct and incorrect labels.

CHAPTER 2:
The Data

The creation of our dataset was a challenging endeavor resulting from the combined efforts of several organizations. The project began as a collaborative work between MBARI and associates from the Department of Operations Research and the Department of Physics from the Naval Postgraduate School (NPS). The original acoustic data were collected by an underwater hydrophone operated by MBARI and the NPS Physics Department as part of the Monterey Accelerated Research System (MARS) offshore cabled observatory. In order to build the project dataset, the acoustic data gathered by the MARS hydrophone (Figure 2.1) were associated with several weeks of AIS data from the USCG Navigation Center. Once the data were properly associated as "vessel" or "nonvessel," they were then converted into a usable format by standardizing the associated spectrogram values.

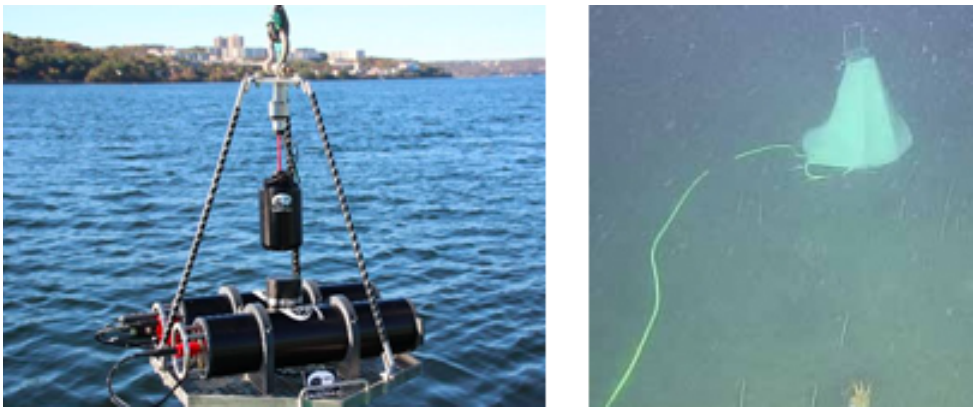


Figure 2.1. Left: MARS hydrophone preparing to submerge. Right: MARS hydrophone fully underwater. Source: Dawe and Ryan (2006).

2.1 Acoustic Data

The acoustic data provided by MARS were collected through an undersea hydrophone located at the bottom of the underwater Monterey Canyon, 30 kilometers off the coast of Central California (Figure 2.2). The hydrophone provides a continuous audio stream of the ambient environment, with the timestamps, acoustic frequencies, and acoustic amplitudes streamed to NPS servers. To date, there are twelve thousand hours of unclassified acoustic

data, occupying more than 5 terabytes. The hydrophone's location within the canyon is ideal because the topography of the area acts as a natural "concert hall" to strengthen any incoming sounds. The hydrophone is able to capture sounds well beyond the limits of normal human hearing. Recorded frequencies range from 10s of Hz to 2000 Hz. These sounds can range from whale calls and dolphin clicks, to weather phenomena such as waves and rain. The focus of this study was the acoustic data produced from ships and vessels, which can transmit the low-frequency sounds of their engines from dozens of kilometers away (Fulton-Bennett 2018).

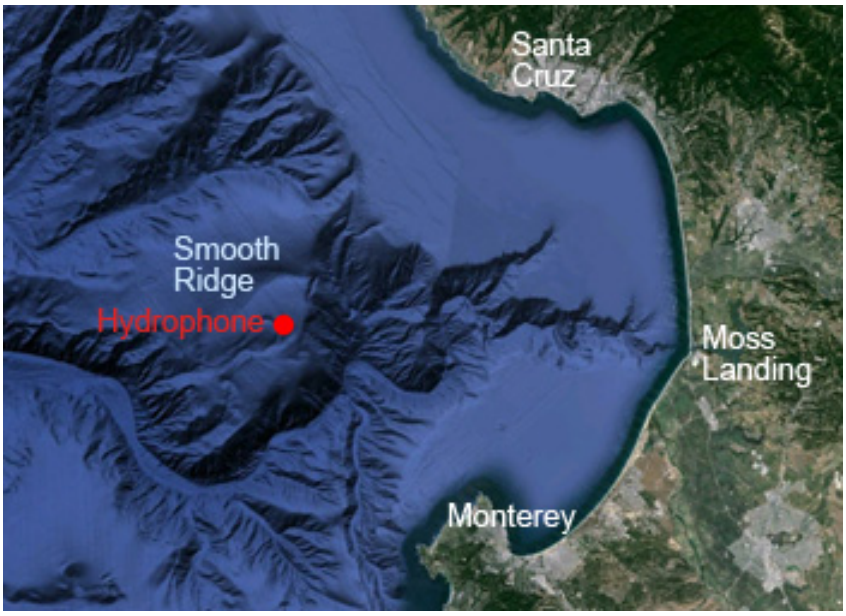


Figure 2.2. Location of the MARS hydrophone. It is lies near a section of the Monterey Canyon dubbed Smooth Ridge. Source: Fulton-Bennett (2018).

Prior to beginning any training of the classification programs used in this study, the recorded acoustic data were converted into spectrogram images. Since the unprocessed acoustic dataset contained time domain-based acoustic information, it was converted into a sinusoidal-based frequency domain by applying a Discrete Fourier Transformation (DFT) in order to more easily interpret the data for analysis. The employment of a DFT allowed the audio signals to be split into small, consecutive segments of length N (Rippel et al. 2015). These pieces were then mapped into a set of N discrete frequency components. Using DFT requires only an order of $N \log(N)$ operations for any given vector within the dataset. This allows the DFT of each N-length segment to be computed relatively quickly, speeding up the

transformation of acoustic data to frequency-based spectrograms for evaluation and analysis (Rippel et al. 2015).

Audio spectrograms are three-dimensional graphical representations of a waveform's frequency and amplitude over time. They are a convenient method for quickly depicting acoustic signal information. Once a signal was transformed via DFT, a discrete frequency index can be obtained to help characterize and classify the signal (Huang 2016).

A common feature present in all of the spectrograms produced by the hydrophone is a frequency band signal of approximately 500 Hz. While the spectrogram is recording, an unintended side effect of its power supply is that the hydrophone's self-noise is also recorded. An example can be seen in Figure 2.3, where amplitude is represented by color, low (dark red) to high (yellow).

2.2 AIS

The MARS hydrophone was able to provide crucial acoustic data, but a method to differentiate times in which vessels were in the vicinity was still needed. To proceed further, the data would need to be labeled accordingly to discriminate which spectrograms contained vessels and which did not. In order to resolve this issue, AIS data from the USCG Navigation Center were analyzed and collaborated with the acoustic data. As shown in Figure 2.4, the Monterey Bay can experience high traffic density, especially from the nearby shipping lanes.

AIS is an automated tracking system for maritime vessels to identify one another and communicate with shore-based establishments. Vessels continuously transmit their individual AIS information, which includes the vessel's identity, position, speed, course, activity, size, tonnage, and several other characteristics. Although primarily used as a collision avoidance system, AIS transmissions are often closely monitored and gathered for data collection and analysis by maritime agencies (USCG Navigation Center 2020a).

The USCG was able to provide AIS data from September and October of 2019 for vessels within the Monterey Bay area. This dataset is comprised of 2,019 different vessels with a total of 4,851,822 AIS transmissions. Several steps were taken to filter the AIS dataset and utilize only the pertinent data points. Because only motor vessels traveling under their own

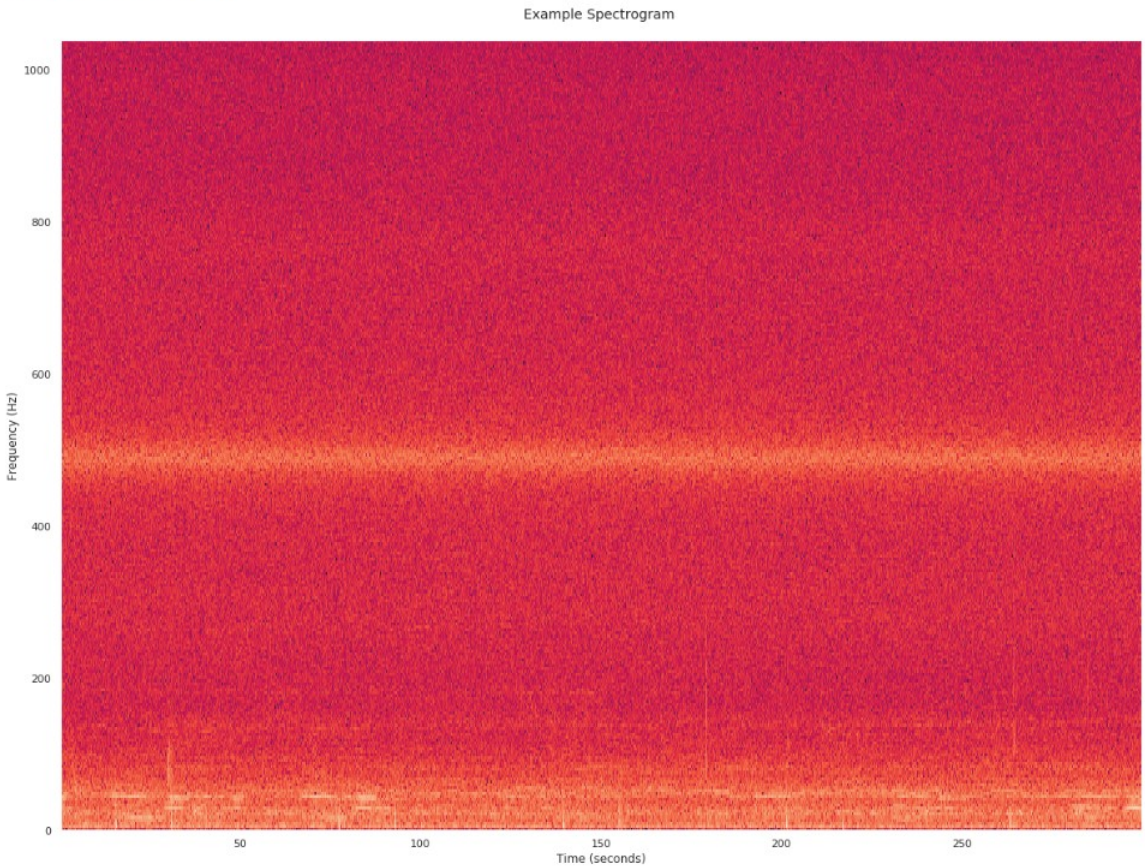


Figure 2.3. A sample spectrogram of 5 minutes, from 0 to 1000 Hz. The frequency band at 500 Hz is due to the hydrophone power supply.

power were relevant, any vessel whose data indicated that they were moored, anchored, adrift, sailing, or fishing were removed from the dataset. Additionally, vessels that were underway but traveling at less than three knots were also eliminated. The location data from these transmissions were then used to determine which vessels were in the vicinity of the hydrophone. Using the coordinates from the dataset, the distance was calculated based on each data point and the location of the MARS hydrophone. The average distance of the vessels from the hydrophone was 117 kilometers, with some as far as over 200 kilometers away. Figure 2.5 shows the variability of distances of the vessels within the dataset. To offset these extreme distances, only time intervals that contained a vessel within 35 kilometers of the MARS sensor array were labeled as containing a vessel.

After the AIS dataset was cleaned, the times in which the vessels were active were asso-

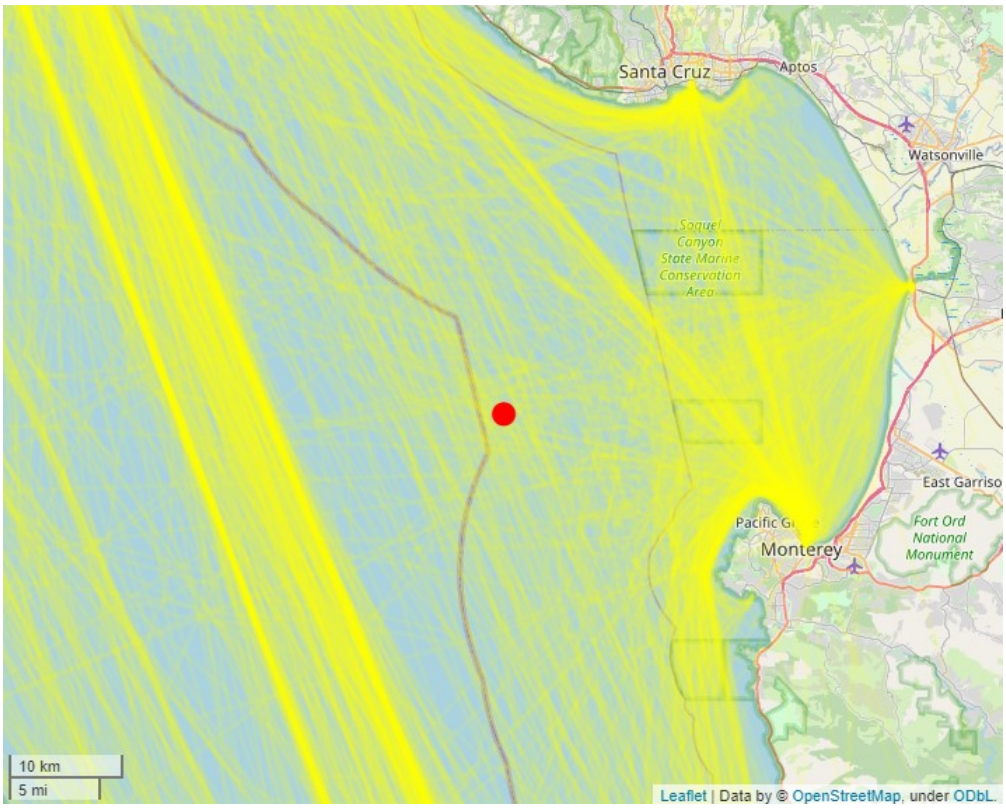


Figure 2.4. A map depicting the Monterey Bay's vessel traffic overall density during September and October of 2019. The red circle signifies the location of the MARS hydrophone. The strong bands to the west indicates a shipping lane.

ciated with the audio data from the hydrophone. Spectrograms from times in which there were no vessels transmitting based off of the cleaned AIS dataset were placed in a "no vessel" category, while times in which one or more vessels were nearby were placed in a "vessel" category. From the spectrograms which had one or more vessels, we were able to visually observe the frequency bands created by vessel engine noise within the lower frequency ranges. The spectrograms without vessels nearby lacked these bands. An example comparison of these spectrograms can be seen in Figure 2.6

2.3 Processing

After the data were sufficiently labeled, they were then transformed and standardized for suitable analysis. To modify the spectrogram data tensors into a useable format, a log

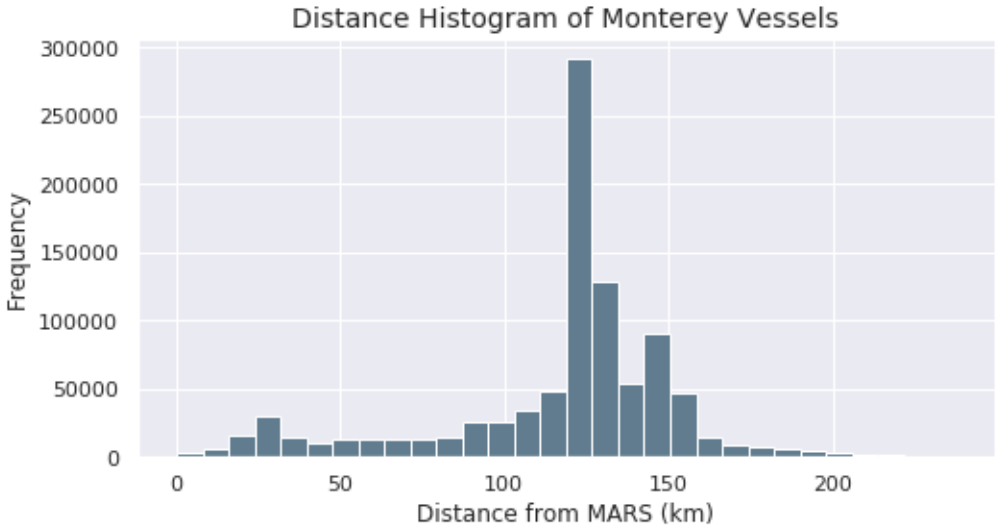
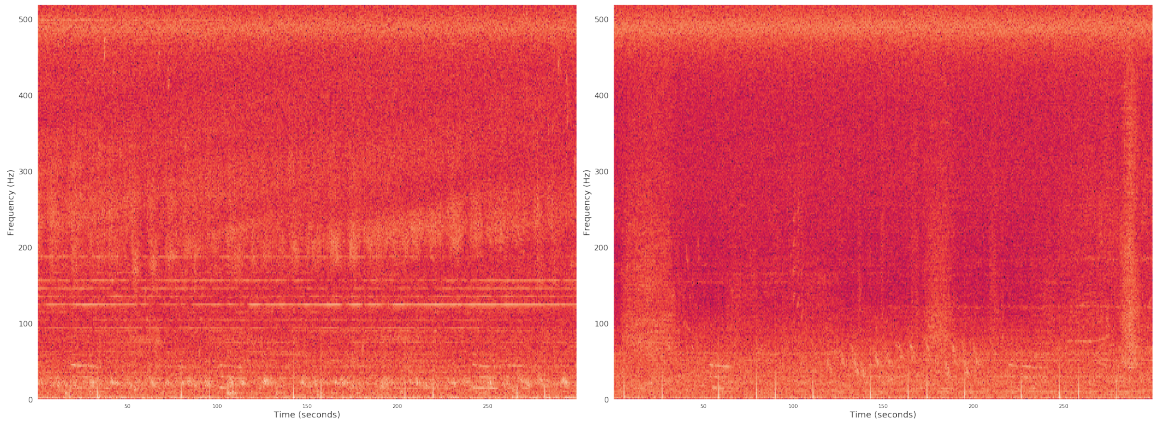


Figure 2.5. Distribution of vessel distance from the MARS hydrophone.

transformation of base ten was applied to each spectrogram's amplitude values (Figure 2.7). We did so to make the distribution of sound intensity more bell shaped, which empirically leads to better performance in classification algorithms. The data were then standardized by dividing the difference between the spectrum tensor values and the overall mean value with the standard deviation. This allowed for a more comparable scale for the wide variety of sounds a classification program may encounter.

Once the data were transformed, we next specified the spectrogram characteristics for the classification algorithms. We divided the acoustic data into 5-minute intervals with frequencies between 0 and 500 Hz. Lower-resolution spectrograms were favored over high-resolution spectrograms due to the amount of unnecessary background noise the high-resolution spectrograms depicted. The overall size of each spectrogram image was specified to be 256x256 pixels. A Python programing library, scipy, was then used to create the final set of spectrogram images. The scipy library's DFT process was inputted with a sampling frequency of 8,000 samples per second and overlapping time segments lengths of 4096. A total of 4,961 spectrograms were created. 20% of these were randomly chosen to be our test dataset, with the remaining going to our training dataset. To expand the pool of available data and provide extra learning, we doubled the size of our training dataset by flipping each spectrogram across the time axis to make a mirrored version of the original images, which



(a) Spectrogram with one or more vessels nearby. (b) Spectrogram with no vessel nearby.

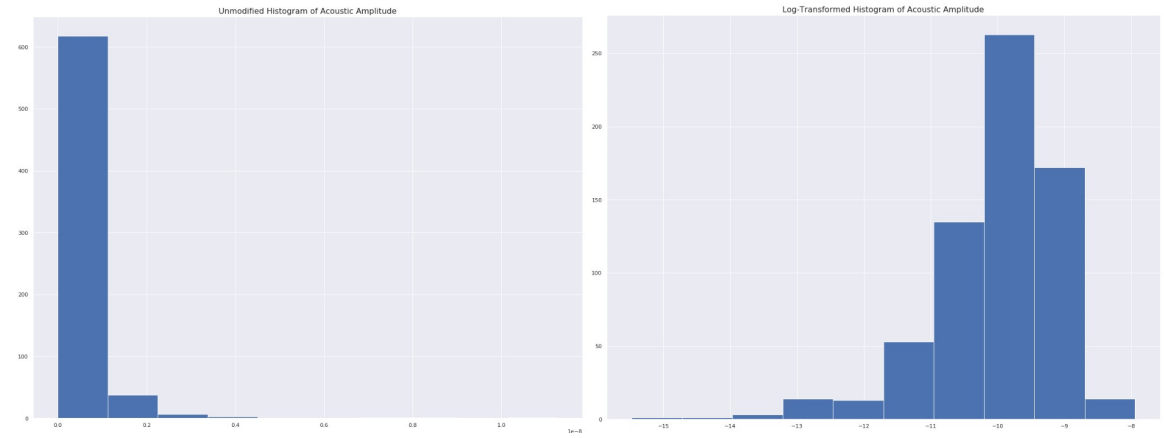
Figure 2.6. Spectrogram comparison of vessel(s) versus no vessels within 35 km, where the vertical and horizontal axes are frequency (Hz) and time (seconds), respectively. Both spectrograms are within a 5-minute time window, from 0 to 500 Hz.

was then added to the training set. This process provided a total of 3,964 additional training spectrograms. A final total of 6,342 training images and 1790 test images was created for our research.

2.4 Assumptions and Data Integrity

Using real-world data always presents challenges and limitations. Stringent efforts were made to clean the data, but a few items were impossible to fully control. One of our primary concerns was that although we had a reasonable degree of confidence in the ground truth of the spectrogram labels, there was an unknown amount of error in the data due to the nature of AIS data. For instance, AIS data is not completely reliable. AIS transmission is only required for certain vessel classes and is otherwise voluntary (USCG Navigation Center 2020b). Furthermore, it is dependent on human operators and as such can be subject to human error (Harati-Mokhtari et al. 2007). Vessels may have transited near the hydrophone without appropriately transmitting their correct AIS information, or may have neglected to transmit any information at all.

We also assumed that loud, faraway vessels would have minimal impact on the learning process, and that the hydrophone self-noise would be a negligible factor. Another factor we



(a) Histogram of original frequency-amplitude values. (b) Histogram of log-transformed values.

Figure 2.7. Distribution of frequency-amplitude values, before and after a log base-10 transformation is applied.

could not completely mitigate was the noises from the local environment, such as weather phenomena and whale calls.

One of the major assumptions we made after the data was cleaned was that the timestamps we had associated were correct throughout the entire period. We noticed occasional timestamp inconsistencies of plus or minus 2 seconds appear sporadically in the acoustic data which we accounted for in our code. In order mitigate our concerns and verify that the timestamps associated with the audio data was accurate, local earthquake data was used for comparison. On October 15, 2019, at 19:42 (coordinated universal time), a 4.7 magnitude earthquake occurred a short distance away from the Monterey Bay (Earthquake Track 2019). When spectrogram data from that time was inspected, a large spike in frequency was clearly visible (Figure 2.8). This spike in frequency helped confirm that the hydrophone's time data was accurate.

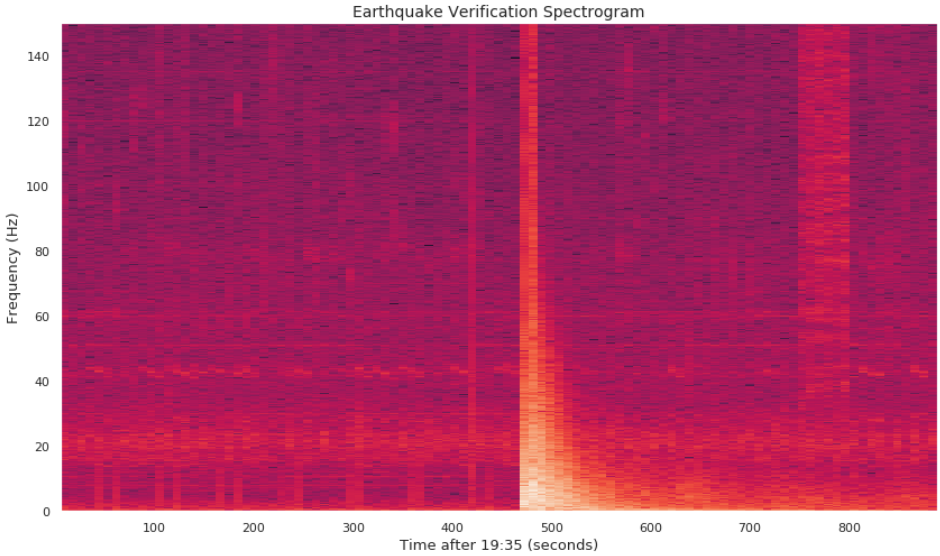


Figure 2.8. Spectrogram depicting an earthquake event that occurred near the Monterey Bay, recorded by the MARS hydrophone.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 3:
Methodology

Several CNN architectures were explored and utilized throughout our research. Pretrained neural networks with histories of success in classifying images were selected to see how well they would perform on our spectrogram dataset.

3.1 What is a Neural Network?

For most of modern history, computers have been immensely helpful tools for automated tasks problem-solving. The types of problems that computers handled were typically grounded in formal, mathematical rules dependent on logical reasoning, which humans typically have difficulty doing without adequate training. Ironically, however, abstract tasks such as visual recognition, which are intuitively simple for humans, have generally been demonstrated to be immensely difficult to implement with computer algorithms. Artificial neural networks, commonly referred to as "neural networks," are an attempt to mimic the human brain's ability to solve abstract problems (Goodfellow et al. 2016).

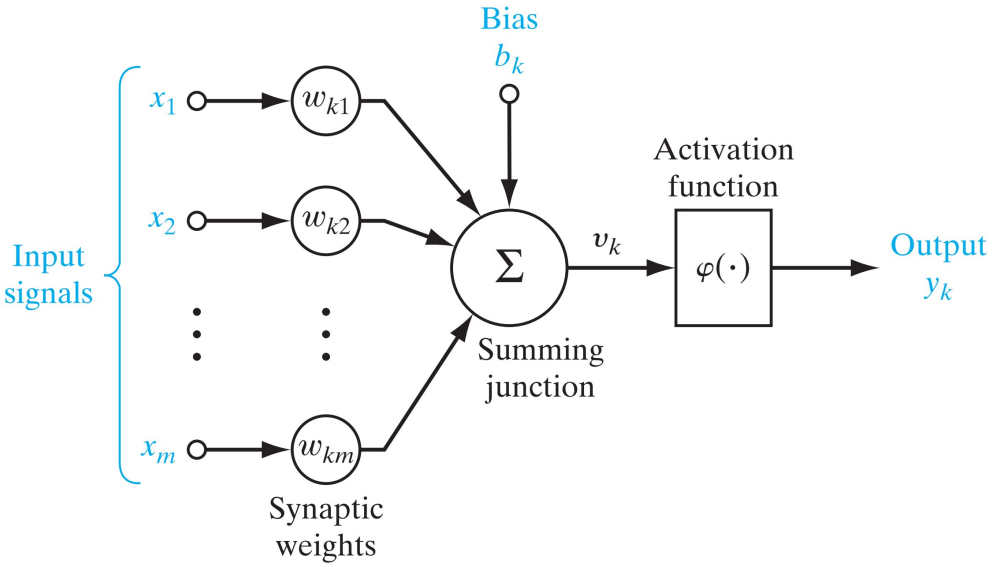


Figure 3.1. A simplified model of a neuron (k) within a neural network. Source: Haykin (2009).

The building block of neural networks are known as neurons, which act as simple processing units for inputted information. Figure 3.1 depicts a diagram of a basic neuron. The interconnection between these neurons creates a massively parallel distributed processor capable of learning characteristics about the input. The strength of the interneuron connections, also known as weights, are used to store the obtained knowledge. To produce an output, a neuron requires at a minimum the inputted values, the weights, and an activation function. An activation function $\varphi(\cdot)$ serves to produce non-linear functions of the inputs. One of the most popular activation function is the rectified linear unit (ReLu) function, $\varphi = max(0, \xi)$ (Li et al. 2020). Additionally, bias is often included as an additional parameter to raise or lower the net output for a more fine-tuned result. The output of a neuron can be mathematically represented by

$$y_k = \varphi(\sum_{j=1}^{m} w_{kj} x_j + b_k) \tag{3.1}$$

where y_k is the output of neuron k, m is the number of inputs, x_j are the input signals, w_{kj} are the weights, and b_k is the bias (Haykin 2009).

Deep neural networks are neural networks in which several layers of neurons are encoded between the initial input and final output. The output of a layer becomes the input of the next. As information passes through these layers, the layer outputs become more and more refined before producing the final result (Goodfellow et al. 2016). Most neural networks are trained using some form of stochastic gradient descent, which is an optimization method based on using first-order derivatives to minimize the error arising from comparing the neural network's final output to the true class of all observations int eh training set. Gradient descent is made "stochastic" by approximating the final gradient using subsets or batches of the training set at each step. To implement stochastic gradient descent on the multiple layers of a deep learning program, the gradients of each layer are computed through recursive application of chain rule (Li et al. 2020). Learning rates should be chosen with care. A learning rate that is too small will generally be more computationally intensive and take longer to converge. On the other hand, larger learning rates are faster but may "overshoot" and end up diverging instead of converging (Bottou 2012).

Our research with spectrograms was primarily based on supervised learning via CNN

architectures. Because the datasets were labeled with classifications beforehand, the learning algorithms within the networks attempted to learn how inputs were associated with the labeled outputs with the given training set (Goodfellow et al. 2016). More details on CNN architectures are outlined in the following section.

3.2 CNNs

CNNs are a powerful form of deep learning based neural networks which specialize in analyzing visual information. To analyze a digitized image, they reduce the image's complexity into a form which is easier to process while retaining features which may be important for classification. Although numerous variations of CNNs exist, they all comprise of the same basic framework (illustrated in Figure 3.2). Like all neural networks, CNNs use neurons as the basis for their structure. The interconnection between neurons within a CNN arise by applying a type of operation known as convolution.

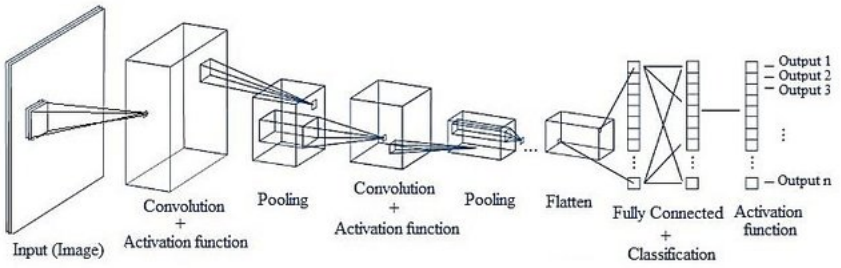


Figure 3.2. The basic structure of convolution neural networks. Source: Hacibeyoglu and Ibrahim (2018).

CNNs are composed of three types of layers: convolutional, pooling, and fully connected layers (Li et al. 2020). These layers are stacked such that the output of a layer serves as the input of the subsequent layer. As the image is processed through the convolutional and pooling layers, the dimensionality (height and width) of the layer shrinks in size. The resulting output, sometimes known as a feature map, is a reduced rendering of the original image in which only the features deemed to be most important are kept. The derived feature maps are then fed into a fully connected layer for classification.

Initializing a CNN requires identification of the number of input channels, the number of output channels, and the *kernel* size for each layer. CNNs analyze the input digitized image

through the lense of a sequence of kernels, sometimes referred to as filters. A kernel in the first convolutional layer acts as a moving window that slides across the digitized image while providing weight values for the convolution (Goodfellow et al. 2016). Figure 3.3 illustrates an example of a 2x2 kernel moving through different portions (often called receptive fields) of a 4x4 single channel image with one value per pixel to form the resulting 3x3 feature map. A colored image is an example of a three channel input image. A single convolutional layer has several such kernels (all of the same size) used to produce an input channel number of feature maps. These then provide the inputs to the next layer.

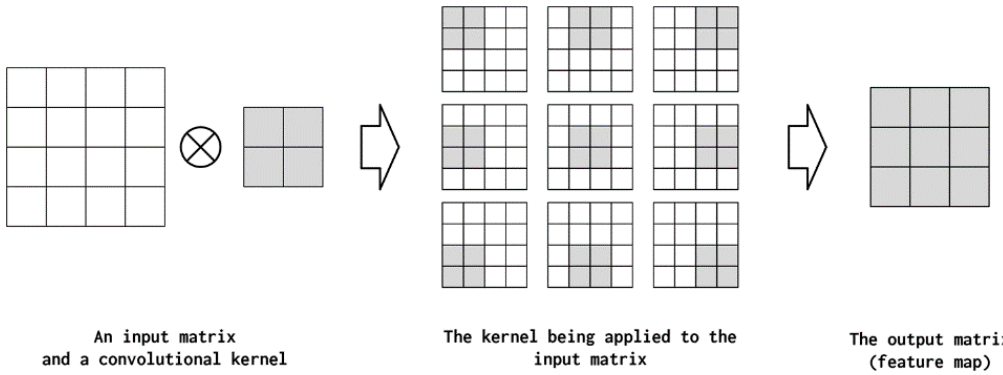


Figure 3.3. A kernel being applied to an image for convolution. Source: Verma (2018).

Convolution is the central component of CNN architecture. Each time a convolution occurs, the weights (and bias) associated with a kernel produce an output which represents an important feature from the receptive fields of the outputs of the previous layer. The outputs in the same feature map are computed using one kernel and its set of weights (Haykin 2009). The output of a neuron in the jth feature map of the lth layer is computed as

$$x_j^l = \varphi\left(\sum_{i \epsilon M_j} x_i^{l-1} w_{ij}^l + b_j^l\right) \tag{3.2}$$

where $\varphi(\cdot)$ is the activation function applied to the convolution, M_j is the set of indices in the receptive field, x_i^{l-1} are the previous layer's outputs in the receptive field, w_{ij}^l are the weights of the jth kernel of the lth layer, and b_j^l is the bias (Alom et al. 2018). The weights and biases of a neural network are the parameters to be estimated. The number of parameters for a convolutional layer is a function of the kernel size, the number of input

channels, and the number of output channels for that layer. The number of parameters for a convolutional layer is determined by the following equations (Li et al. 2020).

$$W = K_{width} K_{height} C_{input} N_K \tag{3.3}$$

$$B = N_K \tag{3.4}$$

$$P = W + B \tag{3.5}$$

where

$$\begin{aligned} W &= \text{Number of weights} \\ K_{width}, K_{height} &= \text{Kernel spatial dimensions} \\ C_{input} &= \text{Number of input channels} \\ N_K &= \text{Number of kernels} \\ B &= \text{Number of biases} \end{aligned}$$

Additional architecture choices for a convolutional layer include stride, which indicates how far the kernel moves when transiting across the previous layer, and padding, which adds an outside border of a specified width to the previous layer so that more details are retained. The dimensions of the outputted feature map (PyTorch 2019) are determined by

$$height_{output} = \frac{height_{input} - K_{height} + 2\rho}{S_{height}} + 1 \tag{3.6}$$

$$width_{output} = \frac{width_{input} - K_{width} + 2\rho}{S_{width}} + 1 \tag{3.7}$$

where

$$\begin{aligned} height_{input} &= \text{Height of previous layer before padding} \\ width_{input} &= \text{Width of previous layer before padding} \\ S_{height} &= \text{Stride height} \\ S_{width} &= \text{Stride width} \\ \rho &= \text{Padding} \end{aligned}$$

As the image is filtered through the image, the weights are held as constant for each kernel. This allows the kernels to reduce the number of parameters and perform transformations of the input for more refined classification (Goodfellow et al. 2016).

Most CNNs also include at least one pooling layer, sometimes referred to as a sub-sampling layer, which reduces the height and width of each convolutional layer even further. Like convolution, pooling involves using a moving window of a specified size to analyze the inputted image. However, pooling performs a statistical function for each component studied, typically either the maximum value or the mean. Pooling layers are useful for determining important features which are not affected by changes in scale or orientation (Goodfellow et al. 2016). Figure 3.4 represents the mechanics of a max pooling layer.

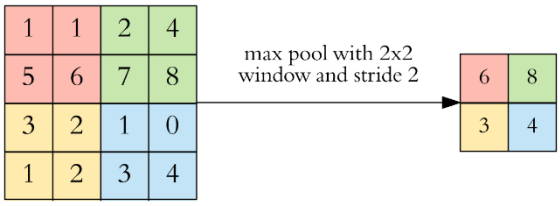


Figure 3.4. A feature map being processed by a pooling layer with a 2x2 kernel size and stride of 2. Source: Verma (2018).

The final stage of CNN architecture are fully connected (FC) layers. The last FC layer assigns a "score" to each classification class based on the outputs of the previous layers. Once the original image has been filtered to a low-resolution representation of the original, it is flattened into a one-dimensional vector which captures the most important features determined by the convolution and pooling layers. The last FC layer interprets this vector to determine a final classification.

In our initial attempts to apply CNNs to the dataset, we built simple networks consisting of only a few layers of convolution and pooling. However, we quickly found that these simple networks were insufficient and were unable to learn from the spectrograms with any reasonable measure of confidence. We turned to pre-existing CNNs with successful reputations to see how they would fare. The general architecture described above were present in the two primary CNNs used in this study, AlexNet and ShuffleNet. Despite the similar structures, the inner mechanisms of the two networks differ in several ways.

3.3 AlexNet

AlexNet is an advanced CNN architecture that rose to prominence following its first-place showing at the 2012 ImageNet Large Scale Visual Recognition Challenge (Alom et al. 2018). Although CNNs had existed previously, AlexNet was a radical breakthrough for artificial image recognition and is considered a major milestone in the history of machine learning. Most CNNs before AlexNet featured only a single convolutional layer followed by a pool layer, but AlexNet's architecture emphasized convolutional depth, with multiple layers stacked on top of each other. Figure 3.5 illustrates a breakdown of the AlexNet's architecture.

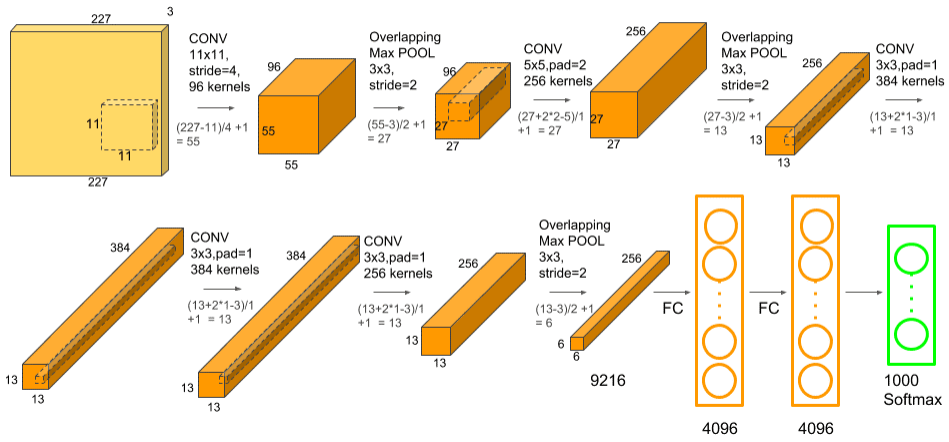


Figure 3.5. The architecture of the original AlexNet convolutional neural network. Source: Hassan (2018).

The ImageNet dataset which was originally used to train AlexNet had over 15 million images of size 227x227 (Li et al. 2020), with 1,000 different possible classifications. The architecture employs five convolutional layers, three max pooling layers, and three FC layers. The initial image is inputted into a convolutional layer with kernel size 11x11, followed by a max pooling layer. Convolution and max pooling repeats again before two convolutional layers are applied in a row. A final max pooling layer feeds the reduced image to three FC layers to obtain a final classification result (Krizhevsky et al. 2012). Full details of these layers can be seen in Table 3.1, where the number of parameters for the convolutional layers are determined by equations 3.3-3.5, and the output channel sizes are computed

using equations 3.6-3.7. The AlexNet output layer, "fc8," originally contained 1,000 output channels, one for each image class, but was replaced by a layer with one output channel whose output is a score where high values represent the presence of one or more vessels, and low values represent no vessels within the vicinity of the hydrophone.

AlexNet's historical success was a major basis for why it was chosen as a focal point of our study. We hoped AlexNet's philosophy of using convolutional depth with additional classification layers would lend itself to be an apt architecture for spectrogram analysis.

Table 3.1. AlexNet Structural Details.

Layer	Stride	Padding	Kernel Size	# Input Channels	# Output Channels	Output Size
conv1	4	0	11x11	3	96	55x55
maxpool1	2	0	3x3	96	96	27x27
conv2	1	2	5x5	96	256	27x27
maxpool2	2	0	3x3	256	256	13x13
conv3	1	1	3x3	256	256	13x13
conv4	1	1	3x3	384	384	13x13
conv5	1	1	3x3	384	384	13x13
maxpool5	2	0	3x3	256	256	6x6
fc6	-	-	1x1	9216	4096	-
fc7	-	-	1x1	4096	4096	-
fc8	-	-	1x1	4096	1	-
Total number of parameters: 18,690						

3.4 ShuffleNet

First introduced in 2017, ShuffleNet is a CNN architecture which was designed for extreme computational efficiency while still being able to produce accurate results. ShuffleNet is an ideal architecture for platforms with limited computational budgets, such as drones, robots, and smartphones. When first implemented on an off-the-shelf mobile device with the 2012 ImageNet dataset, ShuffleNet was able to maintain comparable accuracy with AlexNet while achieving a 1,300% increased speedup time. The two cornerstones of ShuffleNet architecture are pointwise group convolution and channel shuffle (Zhang et al. 2018).

The basic idea of group convolution is that the input is partitioned into groups that are convoluted separately. The outputs are then concatenated. This allows convolutional operations to be distributed to decrease the computational intensity needed. ShuffleNet was designed

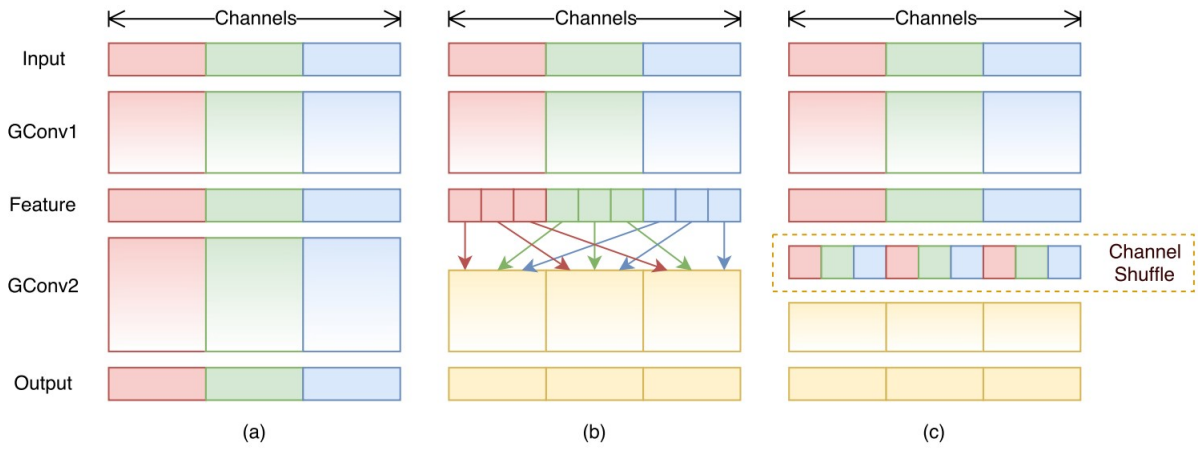


Figure 3.6. Two stacked group convolutions (GConv1 and GConv2) undergoing a channel shuffle. (a) The number of groups remains the same for both stacked convolution layers. (b) GConv2 takes data from different groups, causing the input and output channels to be fully related. (c) Same as (b) but using a channel shuffle implementation. Source: Zhang et al. (2018).

to use 1x1 kernels in its convolutions (known as pointwise convolutions), which allows for greater granularity in feature learning. Additionally, pointwise group convolution apply sparse connections between layers such that each convolution operates only on the associated input channel group (Zhang et al. 2018). This can drastically reduce computational cost. However, a major drawback of group convolution is the when stacked in multiple groups, outputs are effectively "bottlenecked" such that they can only derive information from a small fraction of the input channels, creating weaker classification results (Zhang et al. 2018). Figure 3.6(a) depicts an example of two stacked group convolutions.

To address this problem, ShuffleNet employs a channel shuffle operation to the feature map. When a group layer produces a feature map, the map's output channels for each group are further divided into subgroups. These subgroups are randomized in a channel shuffle and given to the groups of the next layer (Zhang et al. 2018). This process is illustrated in Figure 3.6(b) and (c).

ShuffleNet takes advantage of pointwise group convolution and channel shuffling through stages known as a ShuffleNet unit. The units can be conceptualized as blocks of operations through which the layers are grouped and shuffled for their output. These stages may be repeated several times for each of the output channel groups (Zhang et al. 2018). Table 3.2

displays ShuffleNet's layer architecture in detail.

ShuffleNet's emphasis on computational efficiency coupled with high classification accuracy made it a good candidate for study with the spectrogram dataset. Although our dataset was relatively small, we reasoned that any success found with ShuffleNet could theoretically then be transferred to larger, more complicated datasets.

Table 3.2. ShuffleNet Structural Details.

Layer	**Stride**	**Repeat**	**Kernel Size**	**# Output Channels (g groups)**					**Output Size**
				g=1	**g=2**	**g=3**	**g=4**	**g=8**	
conv1	2	1	3x3	24	24	24	24	24	112x112
maxpool1	2	-	3x3	-	-	-	-	-	56x56
stage2	2	1	-	144	200	240	272	384	28x28
	1	3	-	144	200	240	272	384	28x28
stage3	2	1	-	288	400	480	540	768	14x14
	1	7	-	288	400	480	540	768	14x14
stage4	2	1	-	576	800	960	1088	1536	7x7
	1	3	-	576	800	960	1088	1536	7x7
globalavgpool	-	-	7x7	-	-	-	-	-	1x1
fc	-	-	-	1	1	1	1	1	-
Total number of parameters: 40,508									

3.5 Implementation

We employed the AlexNet and Shufflenet to observe how well they would perform on the spectrogram dataset. For both architectures, two separate methods were used. We first used AlexNet and ShuffleNet such that only the weights (and biases) of the fully connected layers were trained with the dataset. These runs were pre-trained in the sense that the convolutional layer weights were fixed at their original values while the fully connected layer weights were the only parameters estimated with the dataset. Pretrained networks were advantageous for this study because the model could be trained without a massive dataset. At 3,964 images, the dataset was comparatively small for training a neural network. AlexNet and ShuffleNet were then re-trained with all weights estimated from the dataset. The final results exhibit how well each of these approaches fared in performing a binary classification of the spectrograms either containing a vessel or not containing a vessel within the given radius.

From our training dataset, the spectrogram image size of 256x256 was inputted into each of our networks. The networks were initialized with a selected a learning rate of .0003 and batch size of 100 spectrograms to run for a total of 1 million epochs.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 4:
Results

The AlexNet and ShuffleNet architectures produced disappointing classification results. Each network failed to classify with a high degree of confidence whether with either a pre-trained or fully trained architecture. The results are displayed in Table 4.1. A common theme throughout all of the runs was high prediction accuracy for the training set which sharply declined during the test set.

Table 4.1. AlexNet and ShuffleNet End Results

		AlexNet		**ShuffleNet**	
		Pre-trained	Fully Trained	Pre-trained	Fully Trained
Training Accuracy	proportion	0.977	0.972	0.719	0.977
	std. dev.	0.0155	0.0495	0.0557	0.0292
Test Accuracy	proportion	0.507	0.503	0.500	0.499
	std. dev.	0.0233	0.0254	0.0160	0.0154

4.1 AlexNet Results

AlexNet appeared to produce near-perfect prediction accuracy for the training set for both the pre-trained and fully trained models.

As we can see from Figure 4.1, the pre-trained AlexNet architecture was able to predict the vessel labels with astonishingly-high accuracy for the training set, with a correct classification rate of 0.977 and standard deviation of 0.0155. We can also see from that after 13 million iterations the performance stabilizes dramatically. However, the accuracy fell dramatically once the network attempted to predict the test set. The correct classification rate decreased to 0.507 with a standard deviation of 0.0233, making the testing accuracy little better than random guessing.

Figure 4.2 depicts the results of the fully trained AlexNet architecture. The results are similar to the pre-trained AlexNet classification rates in that the training set produced extremely

accurate classification rates but the test set fared much more poorly. We can also see from the graphs that the variance of the classification accuracy is much narrower than that of the pre-trained networks. Both runs of AlexNet indicated that the network severely overtrained on the training set and was unable to generalize the features learned when presented with new data. Following the AlexNet runs, we then proceeded to check the classification ability of ShuffleNet.

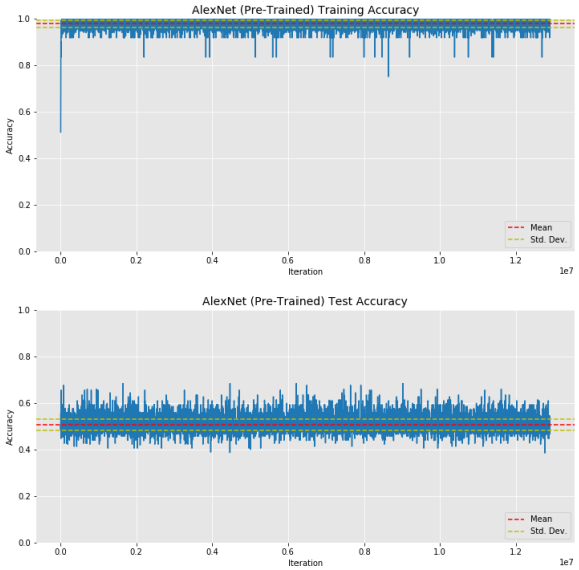


Figure 4.1. Top graph: Training accuracy of the pre-trained AlexNet architecture. Correct classification rate: 0.977, standard deviation: 0.0155. Bottom graph: Testing accuracy of the pre-trained AlexNet architecture. Correct classification rate: 0.507, standard deviation: 0.0233.

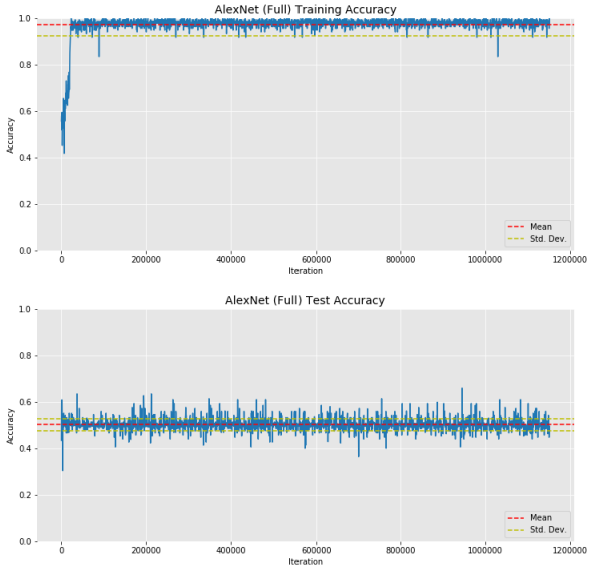


Figure 4.2. Top graph: Training accuracy of the fully trained AlexNet architecture. Correct classification rate: 0.972, standard deviation: 0.0495. Bottom graph: Testing accuracy of the fully trained AlexNet architecture. Correct classification rate: 0.503, standard deviation: 0.0254.

4.2 ShuffleNet Results

In Figure 4.3, we see that the correct classification rate of the pre-trained ShuffleNet architecture was lower than that of either of the AlexNet runs, with a mean classification rate of 0.719. Notably, the pre-trained ShuffleNet classification rate appeared to stabilize at a more gradual rate than that of the AlexNet architectures, flattening out around 0.2×10^6 iterations. Additionally, the number of iterations needed to stabilize the network's feature learning occurred at a more gradual rate. Although the training results do not appear to be overtraining, the test results show that the network is again not able to account for unknown data.

As seen in Figure 4.4, the results of the fully trained ShuffleNet architecture shared the same narrative as the previous runs. Again, the network showed a high level of overtraining for the training set accompanied by a testing accuracy of approximately 50%.

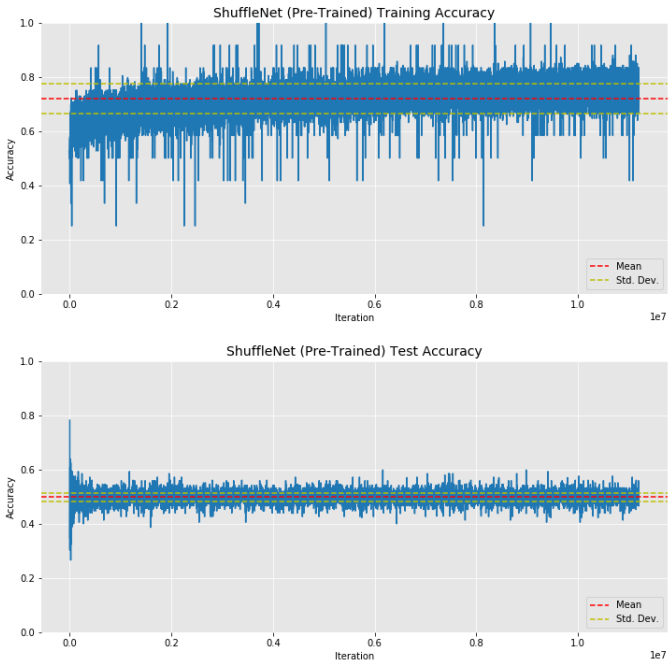


Figure 4.3. Top graph: Training accuracy of the pre-trained ShuffleNet architecture. Correct classification rate: 0.719, standard deviation: 0.0557. Bottom graph: Testing accuracy of the pre-trained ShuffleNet architecture. Correct classification rate: 0.500, standard deviation: 0.0160.

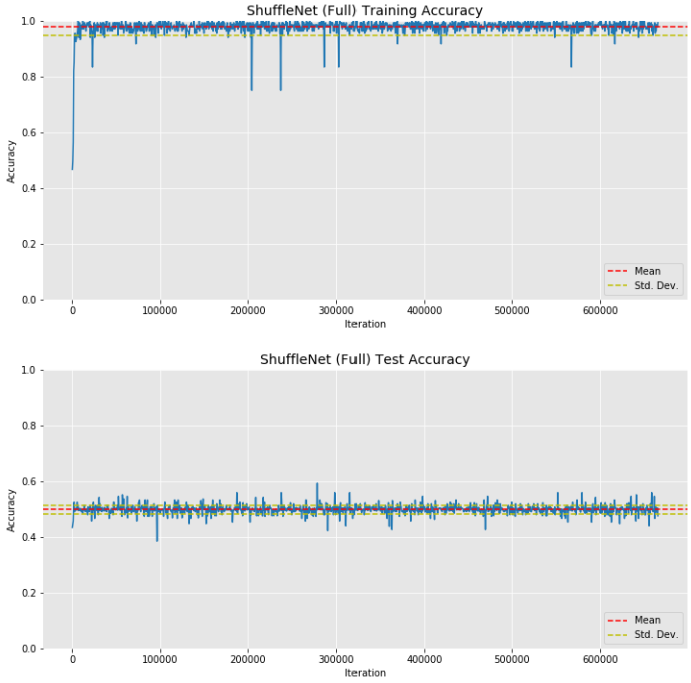


Figure 4.4. Top graph: Training accuracy of the fully trained ShuffleNet architecture. Correct classification rate: 0.977, standard deviation: 0.0292. Bottom graph: Testing accuracy of the fully trained ShuffleNet architecture. Correct classification rate: 0.499, standard deviation: 0.0154.

Despite our efforts, the AlexNet and ShuffleNet architectures were unable to accurately classify the spectrogram test set. The failure of these networks may in part be due to the nature of the dataset. As alluded to within Chapter 2, there were several unknown variables when labeling our dataset. Vessels which passed over the hydrophone but were not emitting AIS may have tainted the "no vessel"-labeled spectrograms in such a manner that the architectures could not choose the essential features needed for prediction. Additionally, natural factors outside of human control, such as environmental noise from weather phenomena or biological wildlife, may have further confused the dataset with undesirable features. Regardless of the reason for the classification inaccuracy, we decided that further experimentation could be had with the dataset using other varieties of machine learning algorithms.

4.3 Additional Analysis: Mean Frequency Vectors

The original premise of our study was to ascertain how well convolutional neural networks would perform for spectrogram image recognition. After the results of our AlexNet and ShuffleNet classification attempts, we decided to re-focus our approach from classifying the spectrograms based on analysis of the entire image to classification based on mean frequency vectors (MFV). For these experiments, we did not utilize the mirrored training set, meaning the algorithms each used a total of 3,964 spectrogram images. The number of test images remained the same.

For each spectrogram within our dataset, the mean frequency was taken across the time axis in order to convert the images into vectors. We reasoned that although converting the images to vectors might cause some information loss, a major advantage was that background noise could be eliminated, thereby providing a classifier with potentially higher quality data. We then applied several different machine learning algorithms to the flattened dataset to see whether MFVs could yield fruitful results: Quadratic Discriminant Analysis (QDA), classification trees, Gaussian Process (GP) classifier, K-nearest neighbors (KNN), support vector machine (SVM), multilayer perceptron (MLP) neural network, and logistic regression. Brief descriptions of these algorithms are detailed below.

4.3.1 Quadratic Discriminant Analysis

We first assessed our newly-vectorized dataset through Quadratic Discriminant Analysis (QDA). This classification method involves using a generative model where the given class (vessel/no vessel) of the MFVs are assumed to be independent and identically distributed according to a multivariate normal distribution, and where class is modeled randomly according to a prior distribution. QDA evaluates parameter estimates and uses Bayes' Theorem to predict the classification based on the class posterior distribution given to the MFV (James et al. 2013).

4.3.2 Classification Tree

Classification trees are a nonparametric classification method for supervised learning which attempts to predict the output classification through simple rules learned from the data features. They are regarded as greedy algorithms which takes a top-down approach to

conduct recursive binary splitting when examining the input data. They divide the predictor space into non-overlapping regions and measure the qualities of observations which fall within a given region in order to produce a classification (James et al. 2013).

Classification trees are sometimes favored due to their easy interpretability but may be unsuitable for more complicated datasets. Using a classification tree also runs a risk of overfitting due to their difficulty generalize data. Additionally, unwanted bias may be added if a given class dominates over the others (James et al. 2013).

4.3.3 Gaussian Process Classifier

A Gaussian Process (GP) classifier is based on using stochastic processes on Gaussian distributions such that all linear combinations of predictor variables follow a multivariate normal distribution. A latent function f is applied to a logit function in order to obtain a probabilistic class. Although the posterior of f is not typically Gaussian for discrete class labels, the likelihood corresponding to the logit is approximated via a Laplace-based Gaussian approximation. GP classifiers are considered a versatile method of classification due to their ability to interpolate observations (Rasmussen and Williams 2006).

4.3.4 K-Nearest Neighbors

K-Nearest Neighbors (KNN) is a simple yet effective non-parametric method of classification. A KNN algorithm predicts a classification label by basing its decision on a predefined number of training samples (k) closest to the current observation. Since KNN is able to "remember" training data attributes found, it excels at non-generalized learning. KNNs can be easily finetuned by applying cross-validation to its single integer parameter, k, allowing the prediction accuracy to improve with more training data (Goldberger et al. 2005). For our own experiments, we chose a k value of 5.

4.3.5 Support Vector Machine

Like many other classifiers, Support Vector Machines (SVM) produce a score from which they assign classification labels. Unlike other classifiers, this score cannot be interpreted as a probability (Goodfellow et al. 2016). SVM scores are linear functions of weights, biases, and the inputs. They perform classification by constructing a hyperplane as a linear decision

boundary which discriminates the class labels based on which side of the hyperplane a new data points fall (Platt 1999).

SVMs can be used to construct nonlinear decision boundaries through the use of the "kernel trick", where inputs are mapped into a high-dimensional feature space in which the hyperplace is then constructed. The kernel trick allows SVMs to produce classifications based on nonlinear transformations of the original input features. One advantage of SVMs is their computational simplicity, allowing them to be computed easily even when the original input features are replaced by a new set of features via the kernel trick (Platt 1999). In our use of the SVM, we use a radial basis function kernel because we found this kernel yields the better performance than constructing a hyperplane using the original input space or a polynomial kernel.

4.3.6 Multilayer Perceptron Neural Network

Multilayer Perceptron (MLP) neural networks are a predecessor of convolutional neural networks and can be considered a quintessential example of deep learning. Like all deep neural networks, MLPs take inputs through several layers of feature mapping before producing an output. (Goodfellow et al. 2016) The primary difference between a MLP and a CNN is that MLPs utilize a fully connected feed-forward structure which does not contain convolutional layers (Stewart 2019).

4.3.7 Logistic Regression

Logistic Regression is a popular model for binary classification (Goodfellow et al. 2016). It uses a maximum likelihood estimation to estimate class probabilities, where two-class responses are modeled as independent Bernoulli random variables. These probabilities are parameterized as a logistic sigmoid function of a linear function of the input variables, whose coefficients (the weights) are estimated via maximum likelihood. The sigmoid function serves to map the linear function of the inputs to (0,1) (James et al. 2013).

4.4 MFV Results

Each of the machine learning algorithms described above were given the spectrogram training and test datasets in the form of MFVs to observe how they compared to the CNN-

Table 4.2. MFV Classification Results

Classifier	Training Acc.	Test Acc.
SVM	99%	81%
KNN	99%	80%
GP	91%	77%
MLP	99%	75%
QDA	85%	73%
Log. Reg.	72%	72%
Classification Tree	90%	63%

based learning. The accuracy results are outlined in Table 4.2.

The best performing method was support vector machine, though K-nearest neighbors was right on its tail. The Gaussian Process Classifier and fully connected neural network performed similarly, with the Gaussian Process classifier having less pronounced overtraining. We see drastic overtraining on all of the models except logistic regression. The fact that the top performing methods have such high levels of overtraining suggests that we have yet to find the perfect model.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 5:
Conclusion

Although the classification results of the CNN architectures was disappointing, several valuable insights were gained. The inadequacy of the deep learning mechanism of CNNs for learning our vessel spectrogram images could have been due to a variety of reasons. Previously published studies investigating CNN architectures and acoustic data normally utilized highly controlled datasets in which outside variables were accounted for, artificially simulated datasets, or datasets with highly distinct, identifiable sounds which could be characterized by limited ranges of time and frequency. In contrast, the audio spectrograms from the MARS dataset was created in conjunction with numerous uncontrollable factors. As mentioned previously, a major limitation of our research was the inability to completely ascertain ground-truth information regarding the vessel data. Our CNN architectures may have been unable to interpret the dataset due to the inadvertent inclusion of images mislabeled as not containing vessel acoustic information. Additionally, other noise factors such as the ambient environment or biological life may have interfered with the CNN's spectrogram analysis.

Researchers interested in continuing our work may wish to explore machine learning algorithms applied to acoustic MFVs. As we saw, several algorithms exhibited promising results which invite further study. Additionally, numerous other CNN architectures exist which can be used to examine our dataset. These architectures can be modified in a number of ways to accommodate spectrogram data.

The search for a dependable method of autonomous vessel detection is a topic with a wealth of possibilities. We hope our research can help inspire future work to develop practical applications of machine learning in acoustic environments.

THIS PAGE INTENTIONALLY LEFT BLANK

List of References

Alom MZ, Taha TM, Yakopcic C, Westberg S, Sidike P, Nasrin MS, Van Esesn BC, Awwal AAS, Asari VK (2018) The history began from alexnet: A comprehensive survey on deep learning approaches. *arXiv* (1803.01164), https://arxiv.org/ftp/arxiv/papers/1803/1803.01164.pdf.

Becker S, Ackermann M, Lapuschkin S, Müller KR, Samek W (2018) Interpreting and explaining deep neural networks for classification of audio signals. *arXiv* (1807.03418), https://arxiv.org/pdf/1807.03418.pdf.

Bottou L (2012) Stochastic gradient descent tricks. Montavon G, Orr G, Müller KR, eds., *Neural networks: Tricks of the trade*, 421–436 (Springer, Redmond, WA).

Dawe C, Ryan J (2006) Directional hydrophone Https://www.mbari.org/at-sea/cabled-observatory/mars-science-experiments/3d-hydrophone/.

Earthquake Track (2019) 4.7 magnitude earthquake 19 km from ridgemark, california, united states. Accessed May 14, 2020, https://earthquaketrack.com/quakes/2019-10-15-19-42-30-utc-4-7-10.

Fulton-Bennett K (2018) Eavesdropping on the deep. *MBARI* (April 24), https://www.mbari.org/hydrophone-stream-release/.

Goldberger J, Hinton GE, Roweis ST, Salakhutdinov RR (2005) Neighbourhood components analysis. *Advances in neural information processing systems*, 513–520, https://cs.nyu.edu/ roweis/papers/ncanips.pdf.

Goodfellow I, Bengio Y, Courville A (2016) *Deep Learning*. http://www.deeplearningbook.org.

Hacibeyoglu M, Ibrahim MH (2018) Human gender prediction on facial mobil images using convolutional neural networks. *International Journal of Intelligent Systems and Applications in Engineering* 6(3), https://doi.org/10.18201/ijisae.2018644778.

Harati-Mokhtari A, Wall A, Brooks P, Wang J (2007) Automatic identification system (ais): Data reliability and human error implications. *Journal of Navigation* 60(3), https://doi.org/10.1017/S0373463307004298.

Hassan MU (2018) Alexnet – imagenet classification with deep convolutional neural networks. *Neurohive* (October 29), https://neurohive.io/en/popular-networks/alexnet-imagenet-classification-with-deep-convolutional-neural-networks/.

Haykin S (2009) *Neural Networks and Learning Machines* (Prentice Hall, Upper Saddle River, NJ).

Huang HC (2016) *Detection and classification of baleen whale foraging calls combining pattern recognition and machine learning techniques*. Master's thesis, Department of Oceanography, Naval Postgraduate School, Monterey, CA, https://calhoun.nps.edu/handle/10945/51720.

James G, Witten D, Hastie T, Tibshirani R (2013) *An introduction to statistical learning*, volume 112 (Springer, New York).

Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 1097–1105, https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

LaGrone S (2019) Analyst: New photos are 'smoking gun' proving iranian involvement in tanker attack. *U.S. Naval Institute* (June 17), https://news.usni.org/2019/06/17/analyst-new-photos-are-smoking-gun-proving-iranian-involvement-in-tanker-attack.

Li FF, Johnson J, Yeung S (2020) Convolutional neural networks: architectures, convolution / pooling layers. Course notes, CS231n: Convolutional Neural Networks for Visual Recognition, Spring Quarter, Department of Computer Science, Stanford University, Stanford, CA. https://cs231n.github.io/convolutional-networks/.

Lichstein KL, Riedel BW, Richman SL (2000) The mackworth clock test: A computerized version. *The Journal of Psychology* 134(2):153–161, https://doi.org/10.1080/00223980009600858.

Majumdar D (2014) U.S. Navy impressed with new Russian attack boat. *U.S. Naval Institute* (October 28), https://news.usni.org/2014/10/28/u-s-navy-impressed-new-russian-attack-boat.

Majumdar D (2018) A new type of Chinese submarine is supposedly breaking records. here's what we know. *National Interest* (August 27), https://nationalinterest.org/blog/buzz/new-type-chinese-submarine-supposedly-breaking-records-heres-what-we-know-29852.

Moore DF (1991) *Passive sonar target recognition using a back-propagating neural network*. Master's thesis, Department of Electrical and Computer Engineering, Naval Postgraduate School, Monterey, CA, https://calhoun.nps.edu/bitstream/handle/10945/30962.

Niu H, Ozanich E, Gerstoft P (2017) Ship localization in santa barbara channel using machine learning classifiers. *The Journal of the Acoustical Society of America* 142(5):EL455–EL460, https://doi.org/10.1121/1.5010064.

Platt J (1999) Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers* 10(3):61–74, http://citeseer.ist.psu.edu/viewdoc/download;jsessionid=9875A219029084239D958DF9C4B08F52?doi=10.1.1.41.1639rep=rep1type=pdf.

PyTorch (2019) CONV2D. Accessed April 14, 2020, https://pytorch.org/docs/master/generated/torch.nn.Conv2d.html.

Rasmussen CE, Williams CK (2006) *Gaussian processes in machine learning*. http://www.gaussianprocess.org/gpml/chapters/RW.pdf.

Rippel O, Snoek J, Adams RP (2015) Spectral representations for convolutional neural networks. *Advances in neural information processing systems*, https://papers.nips.cc/paper/5649-spectral-representations-for-convolutional-neural-networks.pdf.

Salamon J, Bello JP (2017) Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Processing Letters* 24(3):279–283, https://arxiv.org/pdf/1608.04363.pdf.

Stewart M (2019) Simple introduction to convolutional neural networks. *Towards Data Science* (February 26), https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac.

USCG Navigation Center (2020a) Ais overview. Accessed February 14, 2020, https://www.navcen.uscg.gov/?pageName=aismain.

USCG Navigation Center (2020b) Ais requirements. Accessed February 14, 2020, https://www.navcen.uscg.gov/?pageName=AISRequirementsRev.

Verma A (2018) Pytorch basics — intro to cnn. *Towards Data Science* (January), https://towardsdatascience.com/pytorch-basics-how-to-train-your-neural-net-intro-to-cnn-26a14c2ea29.

Wagner DH, Mylander WC (1999) *Naval Operations Analysis* (Annapolis, MD: Naval Institute Press).

Zhang X, Zhou X, Lin M, Sun J (2018) Shufflenet: An extremely efficient convolutional neural network for mobile devices. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6848–6856, https://arxiv.org/pdf/1707.01083.pdf.

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