International Journal of Pattern Recognition and Artificial Intelligence

DeepWhaleNet: Climate Change-aware FFT-based Deep Neural Network for Passive Acoustic Monitoring

Nicholas Rasmussen†, Rodrigue Rizk†, Omera Matoo‡, KC Santosh†

†*Applied AI Research Lab, Department of Computer Science, University of South Dakota, Vermillion, SD 57069*
‡*Department of Biology, University of South Dakota, Vermillion, SD 57069*
nicholas.rasmussen@coyotes.usd.edu, rodrigue.rizk@usd.edu, omera.matoo@usd.edu, kc.santosh@usd.edu

Climate change poses severe risks to the survival of many whale populations, whose habitats and migration patterns are affected by environmental changes. To detect these whales effectively, especially in deep-sea environments, we need to use AI-based techniques to handle the acoustic diversity and variability of different species. However, current methods for whale detection are based on pre- and post-processing steps that reduce their efficiency and generalizability. To address this issue, we present DeepWhaleNet, a novel deep-learning model that automates whale detection in Underwater Passive Acoustic Monitoring datasets. DeepWhaleNet simplifies the detection process by extracting relevant features from raw log-power spectrograms and helps protect these threatened species by supporting conservation efforts. Our model uses a larger short-time Fourier transform as input and a custom ResNet-18 architecture for classification, which enables it to separate whale sounds from noise and capture their temporal and spectral characteristics. We evaluate the performance of DeepWhaleNet and show that it surpasses state-of-the-art methods, achieving an 8.3% improvement in the F-1 score and 21% higher average precision of binary relevance than the baseline method. Moreover, our model demonstrates its versatility and suitability for species-specific retrieval problems through an ablation study on multi-label retrieval problems and a 99.1% recall for Blue Whales.

Keywords: Deep learning, signal processing, bio acoustics, whale detection, climate change

1. Introduction

Whales are among the most magnificent and mysterious creatures on Earth. They play a vital role in the health of the oceans, where they help provide up to 50% of our oxygen, combat climate change, and sustain fish stocks [1,2]. However, whales are also facing unprecedented threats from human activities, such as overfishing, pollution, ship strikes, and climate change, which affect their habitats and migration patterns. To protect these endangered species, we need to monitor and analyze their populations using advanced techniques that can handle the acoustic diversity and variability of different whale sounds. In this paper, we present DeepWhaleNet, a novel deep-learning model that automates whale detection in Underwater Passive Acoustic Monitoring datasets, without requiring nearly as much pre and post-processing. We show that our model outperforms state-of-the-art methods in terms of accuracy and precision, and can also retrieve specific whale species from multi-

label data. We demonstrate the applicability of our method to help conserve Blue and Fin Whales, two of the most threatened baleen whales in the world.

Using hydrophones, UPAM records the acoustic signals of whales and other marine animals in their natural habitats, which can reveal necessary information about their presence, spatial patterns, behavior, sounds, habitat choices, and human influences [3,4]. However, UPAM faces difficulties, as the noise recorded from various sources, such as ships, wind, waves, and other marine life, degrade the quality of the signals and hinder the precise recognition of whale calls by human experts. Therefore, there is a need for an AI-based technique that can automatically detect and classify whales from large-scale, noisy UPAM datasets.

Correlation kernels are one of the most effective conventional methods for extracting whale calls from these large datasets [3]. These kernels scan specific regions of a heavily processed log-power spectrogram, looking for pixels that match the specific whale vocalization pattern they detect. When the features are carefully chosen and tuned, this method can achieve high recall rates of up to 90% for Fin whale detection [5]. However, their performance can deteriorate under different conditions, such as the frequency change of calls from year to year and the variation from population to population [6]. Moreover, some features of these whales' calls can last up to 40-50 seconds [7], a duration that a correlation kernel struggles to capture.

Few studies have applied deep learning methods for large-scale UPAM whale detection. Rasmussen et al. [8] developed a semi-complex recurrent Convolutional Neural Network (rCNN) where the rCNN would segment an entire audio recording into 9-second chunks and feed them to a CNN for classification. This method shows good performance on a relatively small dataset. Miller et al. [9] used a simpler DenseNet architecture with a smaller 4.5-second window and 2-second overlap. This chunking technique gave their model some classification advantages, as they could have two, or more in some cases, opportunities to be correct on a single whale call. Their method performs well compared to the dataset annotations after heavy pre- and post-processing on a large UPAM dataset with hundreds of hours of hold-out test data.

Given the above facts, this study introduces DeepWhaleNet, a specialized convolutional neural network (CNN) that can accurately distinguish between whales and background noise in UPAM data. Our CNN leverages robust preprocessing techniques that enhance the signal-to-noise ratio and preserve whale calls' spectral and temporal features. The proposed CNN demonstrates superior performance over existing state-of-the-art methods regarding detection accuracy and computational efficiency. Furthermore, our CNN exhibits novel versatility, as it can generalize to different tasks and streamline whale detection processes for a wide range of vocalization tasks.

The remainder of this paper is structured as follows. Section 2 describes the biological context of UPAM usage, along with the collected dataset. Next, section 3

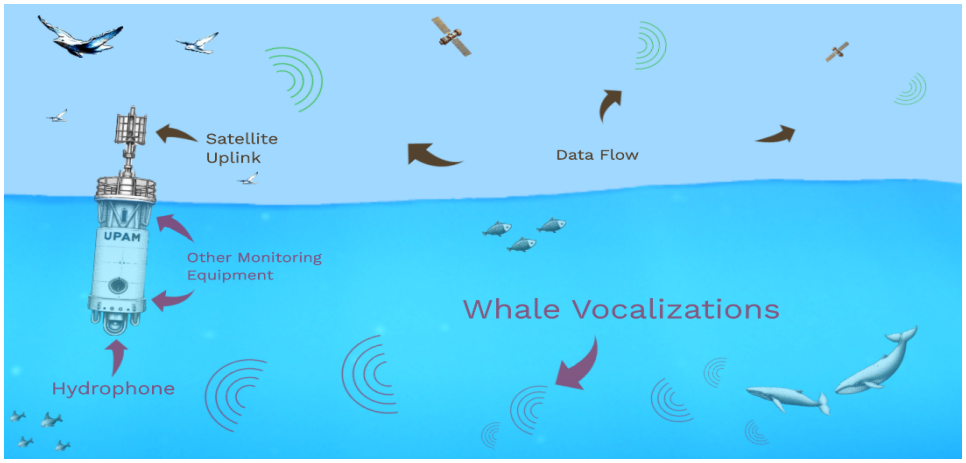


Fig. 1: Illustration of an Underwater Passive Acoustic Monitoring (UPAM) Station: Equipped with a Hydrophone Array, Data Logger, and Satellite Uplink for Real-Time Whale Vocalization Analysis. Other Equipment can include Enviornmental Sensors that record water temperature and salinity. GPS can track non-stationary units.

details the specifics of our proposed model. After, section 4 presents the experiment's testing procedures. Furthermore, section 5 presents the results of our tests. Moreover, section 6 provides a discussion of the findings, and finally, section 7 concludes our paper.

2. Background

This section explores the related work for our experiments. First, our study details the biological context of what our solution offers to whale conservationists, specifically regarding tracking migration patterns. Next, the complex dataset created for UPAM Blue and Fin whale detection along with descriptions of the whale vocalizations are detailed. Last, the selection of Miller et al's [9] method is discussed.

2.1. *Underwater Passive Acoustic Monitoring*

UPAM has been employed in several innovative ways to enhance our understanding of marine species. For instance, the Low-Frequency Detection and Classification System (LFDCS) has been used to detect the calls of marine mammals, including Fins and Humpbacks, by analyzing acoustic data and comparing the attributes of these signals to a reference library of species-specific call types [10]. Furthermore, acoustically assisted sampling designs get used to collect environmental (e)DNA from beaked whales in The Bahamas, with all whales initially located using information from a bottom-mounted acoustic array available on the Atlantic Under-

sea Test and Evaluation Center (AUTEC) range [11]. Moreover, the OPP Whale Detection and Collision Avoidance (WDCA) initiative has focused on evaluating and testing technologies that enable timely detection of whale presence, accurate identification of the species and population, and effective tracking of whale movements [12]. Figure 1 illustrates a diagram of UPAM buoys capturing sounds from whales deep in the ocean.

Blue and Fin whales belong to the baleen whale family and are highly adaptable to various marine habitats, ranging from coastal to oceanic waters. However, centuries of intensive whaling have decimated their populations, leaving Blue whales at less than 1% of their original population and Fin whales at about 10% [13]. This alarming situation calls for urgent and compelling conservation actions to protect these species from extinction. Although some surveys indicate a slight recovery of their populations in recent decades, these data need to be updated and expanded to capture the current status and trends of these whales worldwide [3]. Therefore, applying AI techniques to monitor and analyze their populations could be a powerful tool to evaluate the impact of conservation efforts and identify the main threats and opportunities for their survival. In this paper, the proposed method uses Underwater Passive Acoustic Monitoring (UPAM) and Deep Neural Networks to help protect and conserve Blue and Fin Whales in a changing climate.

Sound is a vital component of the ecology and behavior of whales, especially for communication in the often murky marine environment. Blue and Fin whales produce characteristic vocalizations that can be classified into songs or calls, depending on their structure and function. These sounds are specific to each species and vary geographically and temporally [14, 15]. They serve multiple purposes, such as locating and capturing prey, maintaining social bonds, avoiding predators, and navigating the ocean. UPAM using fixed hydrophones is a cost-effective and non-invasive way to obtain large-scale and long-term data on the acoustic presence and activity of these whales, which can help to track and differentiate them across their habitats and to develop predictive models for climate change adaptation [3,13,14].

One of the challenges of studying Blue and Fin whales is their complex and variable movement patterns. Capital breeders rely on their stored energy reserves to sustain themselves and their offspring, enabling them to separate their feeding and breeding grounds. They typically migrate seasonally between high-latitude feeding areas in summer and low-latitude calving and mating areas in winter. However, not all individuals follow this pattern, as some may exhibit partial or differential migration or feed opportunistically along their migratory routes or in their wintering areas [16]. Various methods can track these movements, such as satellite tags, photo-identification, and this study's method, UPAM.

The previous facts illustrate the versatility and potential of UPAM as a tool in marine biology, allowing researchers to monitor and study marine life in their natural habitats using a non-invasive method. By using deep learning algorithms to

automatically detect and classify whales from noisy and massive UPAM datasets, conservationists hope to provide valuable insights into their occurrence, distribution, behavior, communication, environmental preferences, and human interactions, which can inform and improve conservation strategies for these endangered species. This study aims to evaluate the potential of UPAM and deep neural networks for conserving Blue and Fin whales in the face of climate change.

2.2. *The Antarctic Blue and Fin Whale Acoustic Trends Project*

For investigating trends in Antarctic Blue and Fin Whale population growth, abundance, distribution, seasonal movements, and behavior, the Antarctic Blue and Fin Whale Acoustic Trends Project was started in 2009 [3]. This project has collected about 300,000 hours of audio recording from the Southern Ocean over the last 20 years as of March 2020.

The dataset for testing our proposed method is among the first attempts at a diverse circumpolar Blue and Fin Whale UPAM dataset constructed from the Acoustic Trends Project [3]. It includes data from four Southern Ocean geographic regions, with sites having at least a full year of data from 2014 or 2015 and, ideally, two consecutive years. A single dataset from 2005 was added to extend the temporal span of the library. The data were recorded using various instruments and comprised the 11 different spatial-temporal site years.

A site year is defined as a single UPAM instrument and site with roughly a year's worth of recordings [3], and each site had about 200 hours selected for a subset of the data. A systematic random sampling scheme generates the set of acoustic recordings from the larger site-year dataset. The goal was to obtain a representative sample of the signals recorded considering factors such as SNR, periods of inactivity, and capturing sounds that may produce false positives from a model.

Monthly, 10–18 hours of data were annotated for each site [3]. Furthermore, the sampling scheme also gave the dataset an even annotation distribution through the 24-hour clock cycle. However, four sites employ different sampling schemes, and three had recording duty cycles shorter than an hour. For these sites, the different sampling schemes were applied ensuring a representative sample of the signals recorded. In total, the dataset has 1,880.25 hours of annotated data.

The dataset tracks and monitors different Blue and Fin Whale calls repeated as songs or produced as individual notes unique to their respective species. They use different types of vocalizations, each serving different functions. For instance, the Antarctic Blue Whale produces unit A, a constant frequency tone between 25 and 28 Hz, depending on the year, without other units. Then, unit AB is a unit A tone followed by a partial or complete inter-tonal down sweep (unit B). They also produce a 'z-call' with upper tonal unit A and lower tonal unit C present and downswept unit B either present or absent. Moreover, Blue Whales produce frequency-modulated (FM) calls, also known as D-calls, which are typically longer in duration and lower in frequency than FM calls from Fin and Minke Whales. The other

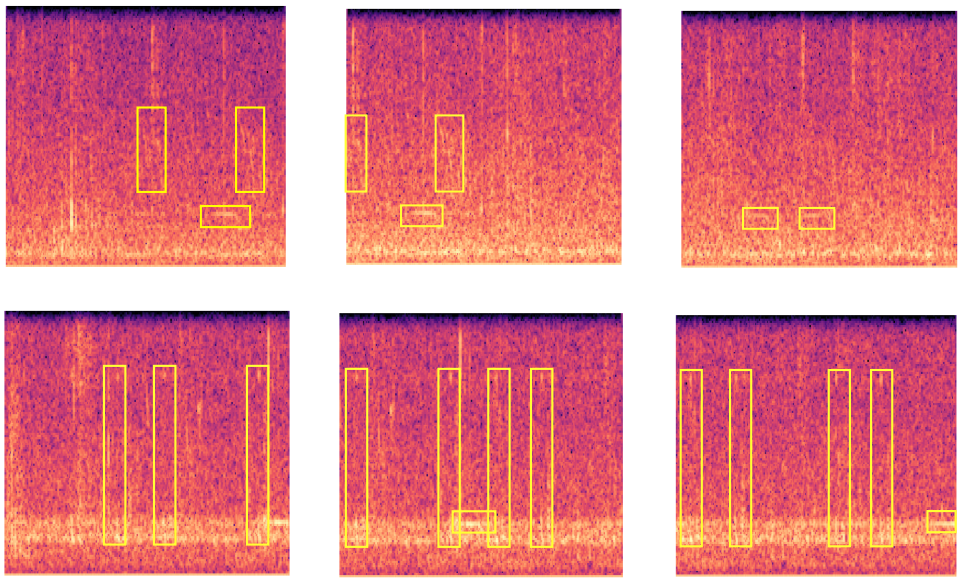


Fig. 2: Unaltered normalized spectrograms with a windowing size of 16,378 and a feature size of $128x256$ from the casey2017 dataset. The top row are Blue Whale vocalizations and the bottom row is Fin Whale vocalizations.

species, Fin Whales, produce a 20 Hz pulse without substantial energy at higher frequencies and a 20 Hz pulse with energy at higher frequencies such as 89 or 99 Hz components. Then, FM calls produced by the Fin Whales are usually downswept and shorter in duration while being slightly higher in frequency than FM calls produced by Blue Whales [3].

One or more of the previous calls can be targeted to create different tasks for machine learning models, either in isolation or in combination. For example, the model can classify a single call type, such as Blue Whale D-Calls, assign a single class to multiple calls from a single species, such as Blue Whale, or group multiple species into a single class, such as all non-target whale calls. Multi-label solutions are also possible, where each sample can contain multiple classes, receiving a probability from zero to one for each class.

2.3. *DenseNet CNN for UPAM Whale Detection*

In large-scale UPAM whale detection, Miller et al.'s method [9] stands out as a benchmark due to its simplicity and effectiveness. They used a DenseNet CNN architecture with a relatively small 4.5-second window and a 2-second overlap. This chunking technique gave their model some classification advantages since they could get two or more chances to be correct on a single whale call. For the

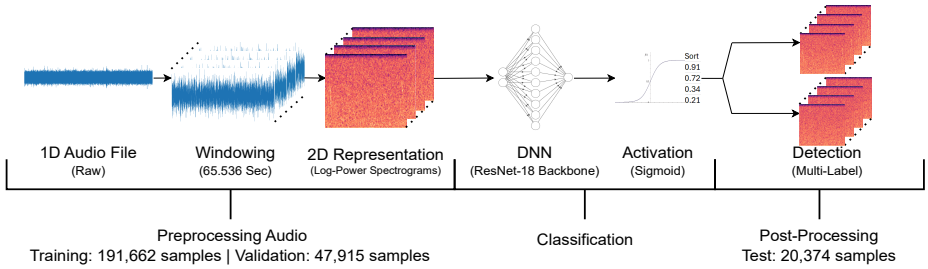


Fig. 3: Schematic diagram of proposed method with 2D data representations (spectogram) as input for the DNN (ResNet-18 backbone) to handle both binary and multi-label tasks.

short-time Fourier transform (STFT), a window of 64 samples and a hop length of 16 were used, resulting in a small spectrogram with a height of 29 and a width of 71 pixels. Some adjustments were made to align more closely with the method described in their paper and to accommodate the specific characteristics of the audio library used in our study. They demonstrated respectable performance versus dataset annotation after heavy pre- and post-processing on a large UPAM dataset with hundreds of hours of hold-out test data.

Our proposed method differs from theirs in several aspects that influence the results. For instance, they discarded all samples with truncated whale calls during preprocessing, ensuring their model would only learn from complete features. In contrast, our method can handle truncated calls without any special treatment, allowing us to increase the number of positive samples in the dataset, which helps to mitigate the imbalance between the classes. Moreover, they down-sampled the negative class to balance the data, a step that may not be necessary in our approach. Heavy down-sampling reduces model robustness by excluding a large portion of informative data. Our model's larger window can capture a more comprehensive representation of the noise vs. signal in the dataset without creating an insurmountable imbalance between the classes. Furthermore, our method did not require tuning the classification threshold for class differentiation, showing that our model is relatively stable at the standard .5 threshold. Thus, tuning this threshold can improve results across all classification tasks. These differences affect performance and will be examined in section 5 and 6.

3. DeepWhaleNet

Previous studies have applied various whale call detection and classification methods, such as signal processing, machine learning, and deep learning techniques. Deep learning has shown promising results in recent years, as it can learn high-level features from raw audio data and handle complex and non-linear relationships between inputs and outputs. However, most existing deep learning models

are designed for specific species or call types of whales and do not generalize well to other scenarios.

This paper proposes DeepWhaleNet, a novel deep learning model that detects and classifies whale calls from different species and call types. DeepWhaleNet is based on a CNN architecture, with additional augmentation layers to improve the robustness and performance of the model. DeepWhaleNet can handle binary and multi-label tasks, and it can adapt to different whale species and call types by fine-tuning the model parameters. To our knowledge, this is the first time a deep learning model has demonstrated such adaptability and generalizability for whale call detection and classification.

3.1. *Method*

Our proposed method, DeepWhaleNet, consists of four main steps: windowing, 2D data representation, the deep neural network, and detection. Figure 3 illustrates the proposed method. Each step is explained in detail.

3.1.1. *Windowing*

The initial phase of our methodology involves the segmentation of unprocessed auditory data into smaller samples suitable for input into our deep neural network. The technique employed for this segmentation is windowing, a prevalent approach in deep learning for the treatment of sequential data modalities, including auditory and textual data. This is achieved by partitioning the original sequence into fixed-size segments [17]. Such segmentation facilitates the efficient management of extensive sequences by the model and the detection of localized patterns within each window.

In the proposed method, 8,192, 16,384, and 32,768 audio vector samples were chosen for the windowing parameter, corresponding to 32.77, 65.54, and 131.07 seconds of auditory data at a sampling frequency of 250 Hz. These window sizes were guided by the observed durations of features from Blue and Fin whales vocalizations, which typically vary from several seconds to approximately 40-50 seconds [7]; like for example, the intervals between Blue Whale three-unit calls, also known as z-calls. Thereby, some of the selected window sizes are designed to contain multiple whale vocalizations and their features, thereby capturing the entirety of their acoustic profiles. Additionally, the choice of window sizes as powers of two serves to enhance computational efficiency while minimizing the potential for data loss.

The windowing technique in this study uses a sliding window with a jump size of half the window size, which means that the window moves by half of its length to create the following sample, resulting in a 50% overlap between adjacent samples. This design choice maximizes the likelihood of capturing entire calls, given their variable lengths and onset times within the audio stream. This overlap is crucial for detecting calls that may straddle the boundary between two windows, ensuring

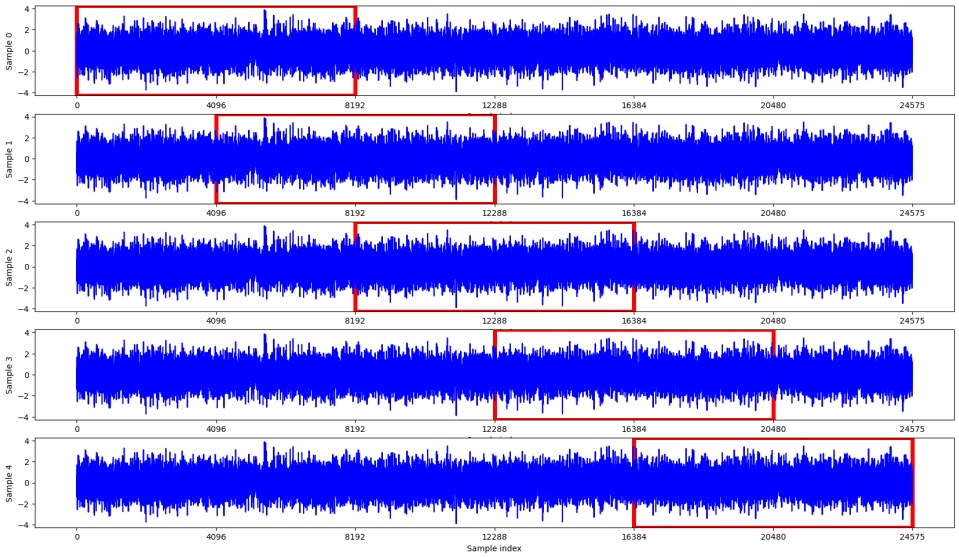


Fig. 4: Illustration of a one-dimensional audio vector with a total length of 24,576 units, segmented into five overlapping windows, each spanning 8,192 units. The highlighted red segment within each waveform indicates the portion of the audio vector selected as a sample. This segment shifts sequentially by half its size to form overlapping samples in the subsequent windows.

that no call is partially represented, and thus potentially misclassified. For example, with a window size of 8,192 audio vector samples, the first window is from element 0 to 8,191 in the audio vector, the second window is from element 4,096 to 12,287 in the audio vector, and so on until the algorithm consumes the entire audio file's vector. The hop size and overlap are chosen to capture all possible whale calls in the audio data and to increase the number of samples for training and testing our model. Figure 4 provides a diagram of the above explanation.

After windowing, each sample gets labels according to the annotations provided by the datasets. For the binary task, a sample is positive if it contains at least one whale call of the target species or call type and negative otherwise. For the multi-label task, a sample gets labels for either or both of the two classes we chose: Blue or Fin whales. Zeros in both class labels represent no target calls from either species. To visualize the effect of windowing on data sample generation, Table 1 summarizes the number of samples and the label distribution for the 11 temporal site years. This distribution offers insights into the prevalence of specific call types within the audio corpus, facilitating a nuanced understanding of the dataset's characteristics.

3.1.2. *2D Data Representation*

The second step of our method is to transform the 1D audio samples into 2D representations that can capture both the frequency and temporal information of the audio signals. The STFT converts the time-domain signals into frequency-domain spectrograms. STFT is defined as follows:

$$\text{STFT}(i, f) = \sum_{k-0}^{L-1} x[bi + k]w[k]e^{-j2(\pi)kf/L}, \tag{1}$$

where b is the hop size, L is the window length, $w[k]$ is the window function, $x[bi + k]$ is the audio signal at time index $bi+k$, and $e^{-j2(\pi)kf/L}$ is the complex exponential function. The STFT computes the Fourier transform of the audio signal within a sliding window of length L and hop size b, resulting in a matrix of complex values representing the signal's magnitude and phase at each time and frequency bin. To begin, values of L and b get chosen to produce spectrograms with dimensions that are powers of two, which generate different matrix shapes and frequency-temporal resolutions of the spectrogram's individual bins. Table 1 shows the values of L and b used for each window size and the features they generate. Feature refers to the size of the feature vector extracted from each windowed segment.

To preprocess the spectrograms the following steps are applied:

1. Take the absolute value of the STFT matrix to obtain the magnitude spectrogram, which represents the signal's energy at each time and frequency bin.
2. Raise the magnitude spectrogram to the power of two to obtain the power spectrogram, representing the signal's power spectrum.
3. Apply the logarithmic decibel (dB) scale to the power spectrogram to obtain the logarithmic power spectrogram, which compresses the signal's dynamic range and makes it more perceptually relevant.
4. Discard the first row and the last column of the log-power spectrogram, as they contain noise and artifacts from the STFT process.
5. Normalize all values between zero and one, as is standard for neural network processing.

The final output of these steps is a log-power spectrogram normalized between zero and one, with dimensions that are powers of two, such as 64 x 512, 128 x 256, or 256 x 128 for the 16,387 audio vector sample windowing technique.

3.1.3. *Deep Neural Network*

The third step of our method is to use a deep neural network (DNN) to learn high-level features from the log-power spectrograms and perform whale call detection and classification. Our base CNN model is a ResNet-18 [18], as the ResNet family has been proven to be a robust and efficient model for audio classification tasks [19]. ResNet-18 comprises 18 layers, including four residual blocks that use skip

Table 1: Class data distributions are presented along with feature engineering parameters. Here, the Window parameter is length of one audio segment, which influences granularity of temporal analysis. L is length of STFT window, determining frequency resolution. b represents hop length, dictating the overlap between consecutive windows and thereby affecting the redundancy and smoothness of the STFT. The columns labeled Total, None, Blue, Fin, and B & F are sample counts for the entire dataset: those without any calls, with only Blue calls, with only Fin calls, and with both Blue and Fin calls, respectively.

Window	**L**	**b**	**Feature**	**Total**	**None**	**Blue**	**Fin**	**B & F**
8,192	128	32	64×256	521,432	434,360	65,389	14,902	6,781
8,192	256	64	128×128	521,432	434,360	65,389	14,902	6,781
8,192	512	128	256×64	521,432	434,360	65,389	14,902	6,781
16,384	128	32	64×512	259,951	202,377	44,678	7,364	5,532
16,384	256	64	128×256	259,951	202,377	44,678	7,364	5,532
16,384	512	128	256×128	259,951	202,377	44,678	7,364	5,532
32,768	256	64	128×512	123,673	90,036	26,549	3,404	3,684
32,768	512	128	256×256	123,673	90,036	26,549	3,404	3,684
32,768	1024	256	512×128	123,673	90,036	26,549	3,404	3,684

connections to avoid the vanishing gradient problem and improve the information flow. Usually, ResNet-18s accept input data with dimensions of 224 x 224 x 3, where 3 is the number of channels. However, our log-power spectrograms have different dimensions and only one channel. Therefore, the ResNet-18 receives the following modifications:

1. Add augmentation layers before the first convolutional layer of the ResNet-18, such as Gaussian noise, horizontal flipping, and random translation. These layers can introduce randomness and diversity to the input data and improve the robustness and performance of the model.
2. Modify the input of the ResNet-18 to accept non-standard input and eliminate the zero padding.
3. Modify all layers to use "same" padding, where the layer's output has feature sizes identical to the input.

Due to performance considerations and the vastness of the dataset, these ResNet-18s get trained from scratch without pre-trained weights, allowing the model to learn the features specific to whale calls [20, 21]. Using the binary cross-entropy loss function for binary and multi-label tasks, each class has its own sigmoid activation. Applying the Adam optimizer [22] with an initial learning rate of 0.01 and a decay of 0.5 every five epochs, the model weights achieving the best validation accuracy for binary tasks and binary accuracy for multi-label tasks during training are selected.

The final part of our neural network uses a sigmoid activation function to obtain the output probabilities for each class in each sample. The sigmoid activation function is defined as follows:

$$f(x) = \frac{1}{1 + e^{-x[i]}}, \tag{2}$$

where e is Euler's number, x is our network's output, and $[\mathrm{i}]$ is the respective node for each class. Here, the sigmoid activation function maps both the Blue and Fin Whale neural nodes to a value between 0 and 1 that both can be interpreted as a probability. The sigmoid activation function has a characteristic S-shaped curve, as shown in Figure 3. This function is monotonically increasing, has a non-negative derivative at each point, and is continuous and differentiable everywhere in its domain [23].

Our proposed method adds an extra dense layer with 256 nodes after the final pooling layer and passes information through the last two dense layers using the sigmoid activation function. The final dense layer has several nodes corresponding to the number of classes in the model's task, thereby obtaining the desired output probabilities for each sample by applying the sigmoid activation function to each node. This allows us to classify as single spectrogram as nothing or only Blue Whale or only Fin Whale or both Blue and Fin Whale for the multi-label problem, since more than one class can appear in any single spectrogram.

3.1.4. *Detection*

The final step of our method is to apply our trained model to the casey2017 dataset [3], which is the hold-out test set. The casey2017 dataset contains Blue and Fin Whale calls from different sites and seasons, background noise, and other marine sounds. The curators annotate the dataset with each whale call's start and end times and the call type. Table 3 and 4 shows the number of positive annotations and negative samples for each task and model parameters.

Using the same windowing and 2D representation steps as described before, the casey2017 dataset is preprocessed. The model predicts the output probabilities for each class in each sample. A threshold of 0.5 is used to determine whether a sample belongs to a class. For example, in the multi-label task, a sample with output probabilities of $[0.8, 0.3]$ is assigned to the first class (Blue Whale), as it has a probability greater than 0.5. The sample is not assigned to the second class (Fin Whale), as it has a probability of less than 0.5. Due to the sigmoid activation, the output probabilities can add up to greater than 1, as a sample can belong to multiple classes.

4. Experiments

This section presents and discusses the experimental evaluation. First, the paper reiterates our selection of the baseline model. Next, the paper explains the evaluation

metrics that measure the performance of the baseline and the proposed models. Then, our paper will examine our method's effect on the evaluation set size. Also, we give a brief explanation of thr pre-trained architectures used for comparison.

4.1. *Baseline Model*

In the preceding sections, we discussed applying deep learning methods to large-scale UPAM whale detection, highlighting the work of Rasmussen et al. [8] and Miller et al. [9] as significant contributions to the field. Our evaluation employs Miller et al.'s method as a baseline model due to its effectiveness and simplicity. Their approach, utilizing a DenseNet CNN architecture with a 4.5-second window and 2-second overlap, is a benchmark against which we compare our proposed method. As outlined in section 3, we have adapted and extended this baseline model to better suit the unique characteristics of the whale calls in the data, aiming to address the limitations observed in previous studies. Our method's ability to handle truncated whale calls and its avoidance of heavy down-sampling of the negative class are key differentiators that we believe enhance the robustness and accuracy of whale call detection. These modifications and our model's performance without threshold tuning are detailed in sections 3 and 5 respectively.

4.2. *Evaluation Metrics*

Four metrics evaluate the performance of DeepWhaleNet and the baseline models: precision, recall, F1-score, and average precision (AP). Our study follows the same evaluation protocol that Miller et al. [9] used for the binary task, which involves concatenating adjacent positive samples for precision, recall, and F1-score. The concatenation process is done because one whale call can span up to three adjacent samples, depending on the window size. The concatenation process affects the precision score, which may include multiple false positives in the concatenated samples. The metrics after the concatenation process are defined as follows:

$$\text{Precision} = \frac{\text{Annotation Hits}}{\text{Annotation Hits} + \text{False Positives}}, \tag{3}$$

$$\text{Recall} = \frac{\text{Annotation Hits}}{\text{Total Annotations}}, \text{ and} \tag{4}$$

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{5}$$

where 'Annotation Hits' are the number of annotated segments captured by a positive sample, 'Total Annotations' are the total number of ground truth positive annotations for a task, and 'False Positives' are the number of concatenated false positive samples. Due to the large window size, our method restricts concatenation to at most three adjacent samples for our proposed model to limit introducing bias into the results, in contrast to Miller et al.'s [9] unlimited concatenation schema.

Table 2: Our proposed method's results in terms of Precision, Recall, F1, AP, number of positive annotations, and number of negative samples.

Task	Prec	Recall	F1	AP	Pos N	Neg N
Blue Only	**.684**	**.991**	**.809**	**.817**	2,971	15,670
MLbl Blue	**.684**	.968	.802	.798	2,971	15,670
Fin Only	.542	.928	.721	.249	293	20,267
MLbl Fin	**.823**	**.932**	**.836**	**.490**	293	20,267

In addition to the metrics used by Miller et al. [9], our study uses the AP metric based on constant interpolation Area Under the Precision and Recall Curve. AP is more robust and informative than the F1-score, as it considers all possible thresholds and trade-offs between precision and recall. The AP metric is also insensitive to the concatenation process. The AP metric is defined as follows:

$$AP = \frac{\sum P_i}{\text{Relevant Samples}}, \tag{6}$$

where P_i is the precision@K for index i, sorted by model probability score, for all relevant samples in the data, and 'Relevant Samples' are the number of relevant samples in the collection. A sample is determined to be relevant if it has a vocalization the model is searching for in the UPAM dataset.

4.3. *Evaluation Set Size*

DeepWhaleNet employs windowing techniques to divide any file into smaller segments, making the model robust to the variation in file length. Tables 3 and 4 show the total dataset size and the number of annotations for each task. The number of negative samples varies across different tasks, but the number of annotations remains constant. Negative samples are the windows that do not contain any whale sounds, while positive annotations are the signals our model tries to retrieve in any positive sample. Therefore, we must account for the trade-off between precision and the proportion of samples generated by the windowing technique. A high proportion of samples can reduce the precision of the model, as it increases the chance of false positive predictions. On the other hand, a low proportion of samples can limit the diversity and coverage of the training data due to each sample being less precise in time. Thus, finding an optimal balance between precision and the number of samples for each task becomes necessary.

4.4. *Pre-trained Architectures*

Our study also includes an assessment of a pretrained ResNet-50 architecture with 'imagenet' weights, compared to a similar setup without 'imagenet' weights. These are referred to as 'Pre-ResNet-50' and 'Scr-ResNet-50' in Table 3. Both used the same

Table 3: Comparative analysis results in terms of Window Size, Feature Size, Precision, recall, F1, AP, number of positive annotations, and number of negative samples for our proposed method.

Task	Window	Feature	Prec	Recall	F1	AP	Pos N	Neg N
Miller et al. [9]	1125	29×71	.428	.677	.524	.304	553	267,384
ResNet-18	1125	29×71	.448	**.703**	.547	.327	553	267,384
Scr-ResNet-50	16,384	$224 \times 224 \times 3$	.167	.429	.240	.175	553	19303
Pre-ResNet-50	16,384	$224 \times 224 \times 3$	.347	.620	.445	.358	553	19303
Proposed Method	16,384	128×256	**.563**	.635	**.597**	**.514**	553	19303

feature generation process as our proposed method, but with an added step of resizing the input to $224 \times 224 \times 3$ to fit the pretrained ResNet-50 model requirements. The pretrained model also demanded fine-tuning of the learning rate to prevent significant performance degradation.

5. Results

This section presents our study's results. First, an analysis of our original results compares the binary and multi-label tasks. After, our paper presents the comparative analysis of the baseline models, the pre-trained models, and DeepWhaleNet. Lastly, our study looks into the effects of different components on model performance in the ablation studies, specifically, our method's windowing technique and feature extraction parameters.

5.1. *Our Results*

In reference to Table 2, the proposed methods for "Blue Only" and "Fin Only" tasks achieved high recall scores of .991 and .928, respectively, indicating that the model is good at identifying positive cases in binary tasks. However, the precision for the "Fin Only" task is relatively low $(.542)$, suggesting that there are 147 concatenated false positives, which is less than a .01 false positive rate, considering all negative samples. The Fin Whale tasks significantly improved performance with the multi-label model, producing a 37.1% increase in AP. However, the opposite was true for the "Blue Only" task, where the binary classification task performed better than the multi-label problem. Some of these phenomena could be explained by confusion between the two frequency-modulated down sweeps that Blue and Fin whales produce and annotation inconsistencies between the two calls.

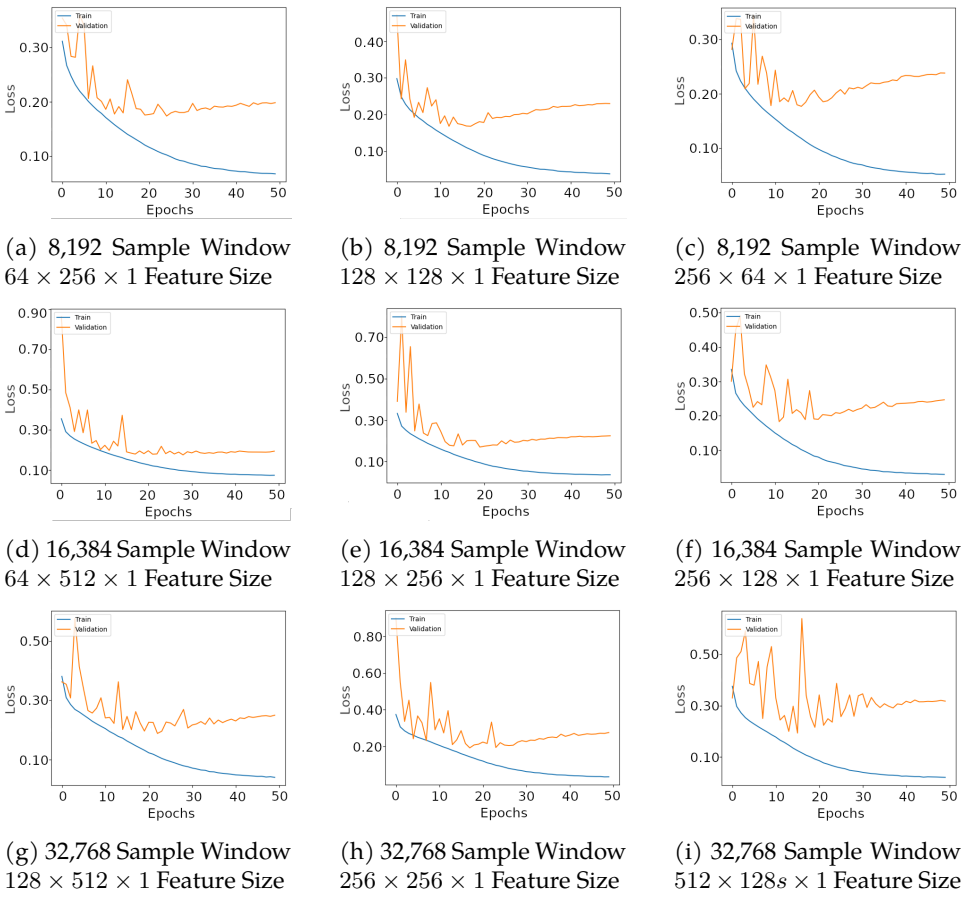
5.2. *Comparative Analysis*

Compared with the DenseNet and ResNet-18 models used with Miller et al.'s [9] window size for the classification of Blue whale D-Calls, our model outperformed

Table 4: Ablation studies' results in terms of Window Size, Feature Size, Precision, recall, F1, AP, number of positive annotations, and number of negative samples for our proposed method.

Task	Window	Feature	Prec	Recall	F1	AP	Pos N	Neg N
MLbl Blue	8,192	64×256	.437	.973	.603	.714	2,971	34,741
MLbl Blue	8,192	128×128	.449	.984	.617	.713	2,971	34,741
MLbl Blue	8,192	256×64	.416	.979	.584	.703	2,971	34,741
MLbl Blue	16,384	64×512	.640	.960	.768	.786	2,971	15,670
MLbl Blue	16,384	128×256	.634	.968	.766	.765	2,971	15,670
MLbl Blue	16,384	256×128	.563	.978	.715	.745	2,971	15,670
MLbl Blue	32,768	128×512	.699	**.989**	.819	**.815**	2,971	6,928
MLbl Blue	32,768	256×256	**.720**	.987	**.832**	.798	2,971	6,928
MLbl Blue	32,768	512×128	**.720**	.984	**.832**	.776	2,971	6,928
MLbl Fin	8,192	64×256	.317	.760	.448	.223	293	40,741
MLbl Fin	8,192	128×128	.385	.808	.522	.251	293	40,741
MLbl Fin	8,192	256×64	.318	.661	.429	.127	293	40,741
MLbl Fin	16,384	64×512	.496	.575	.532	.082	293	20,267
MLbl Fin	16,384	128×256	.515	**.962**	.671	**.620**	293	20,267
MLbl Fin	16,384	256×128	.505	.688	.538	.121	293	20,267
MLbl Fin	32,768	128×512	**.638**	.938	**.791**	.341	293	10,032
MLbl Fin	32,768	256×256	.333	.531	.399	.028	293	10,032
MLbl Fin	32,768	512×128	.194	.151	.170	.015	293	10,032

the baseline models by a large margin, which indicates that our model can learn the features and patterns of blue whale D-calls more effectively than the other models. Moreover, the pre-trained architecture, matched the performance of Miller et al's method regarding AP, however could not match it's performance regarding the other metrics at a .5 decision threshold, suggesting that the threshold needs to be tuned. Although the precision of DeepWhaleNet (.563) is higher than that of both Miller et al. [9] (.428) and ResNet-18 (.448), its recall (.635) is lower than both Miller et al. [9] (.677) and ResNet-18 (.703), as shown by Table 3. These results suggest that while DeepWhaleNet is better at avoiding false positives, it might miss some positive cases. In addition, DeepWhaleNet shows a significant performance increase in AP by 21% over the baseline method; this suggests that the model can benefit significantly from threshold tuning, increasing precision substantially. DeepWhaleNet's architecture also mitigates the significant data imbalance by a ratio of approximately 14:1, as indicated by Table 3.



(a) 8,192 Sample Window $64 \times 256 \times 1$ Feature Size

(b) 8,192 Sample Window $128 \times 128 \times 1$ Feature Size

(c) 8,192 Sample Window $256 \times 64 \times 1$ Feature Size

(d) 16,384 Sample Window $64 \times 512 \times 1$ Feature Size

(e) 16,384 Sample Window $128 \times 256 \times 1$ Feature Size

(f) 16,384 Sample Window $256 \times 128 \times 1$ Feature Size

(g) 32,768 Sample Window $128 \times 512 \times 1$ Feature Size

(h) 32,768 Sample Window $256 \times 256 \times 1$ Feature Size

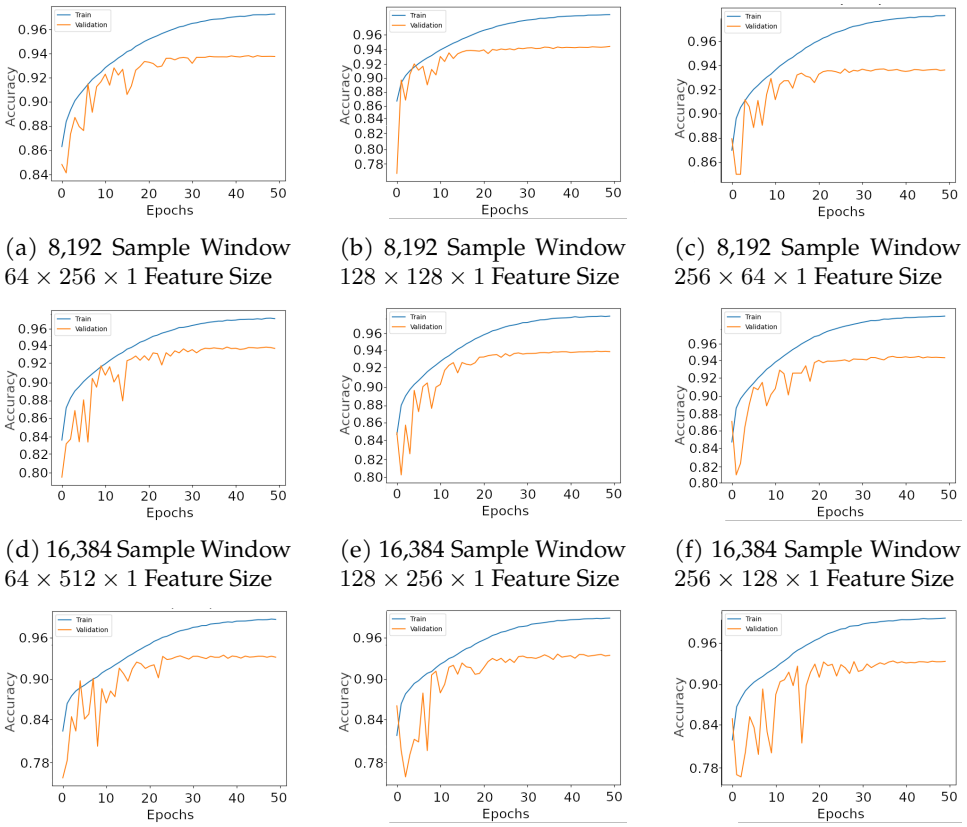
(i) 32,768 Sample Window $512 \times 128s \times 1$ Feature Size

Fig. 5: The proposed model's loss curves trained with different window and feature sizes. The window size is the number of audio vector samples used as input for the model, and the feature size is the size of the spectrogram extracted from each vector.

5.3. *Ablation Studies*

The ablation studies provide insights into how different configurations affect the model's performance. Our methods vary the window size and the feature size of the input data and measure the performance of our model on the multi-label task. The window size determines the duration of each separate sample, and the feature size determines the shape and resolution of the log-power spectrograms. Table 4 correlates the different methods configurations in the ablation studies to their performance on the casey2017 dataset.

For the multi-label Blue Whale task, or "MLbl Blue," the model achieved the highest F1 score (.832) with a window size of 32,768 and feature size of 256×256



(a) 8,192 Sample Window $64 \times 256 \times 1$ Feature Size

(b) 8,192 Sample Window $128 \times 128 \times 1$ Feature Size

(c) 8,192 Sample Window $256 \times 64 \times 1$ Feature Size

(d) 16,384 Sample Window $64 \times 512 \times 1$ Feature Size

(e) 16,384 Sample Window $128 \times 256 \times 1$ Feature Size

(f) 16,384 Sample Window $256 \times 128 \times 1$ Feature Size

(g) 32,768 Sample Window $128 \times 512 \times 1$ Feature Size

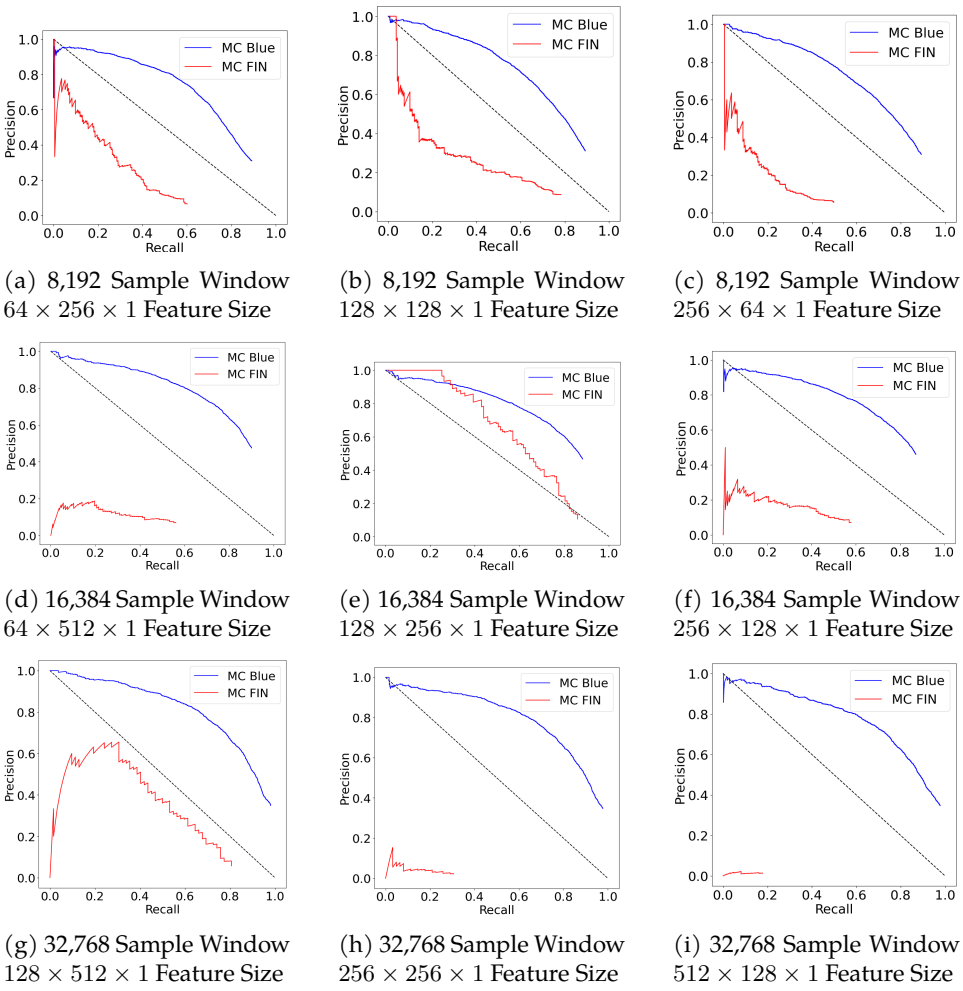
(h) 32,768 Sample Window $256 \times 256 \times 1$ Feature Size

(i) 32,768 Sample Window $512 \times 128 \times 1$ Feature Size

Fig. 6: The Proposed model's accuracy curves trained with different window and feature sizes. The window size is the number of audio vector samples used as input for the model, and the feature size is the size of the spectrogram extracted from each vector.

and 512×128. For the multi-label Fin Whale, or "MLbl Fin" task, the highest F1 score (.791) was achieved with the same window size but a feature size of 128×512. These results suggest that larger window sizes and feature sizes can improve performance.

The larger window size also means concatenating larger portions of time for adjacent positive samples, reducing our detector's temporal specificity. Therefore, our study considers the AP metric, which is unaffected by the concatenation process. For the AP metric, the model achieved the best results with a window size of 16,384 and a feature size of $128x256$ due to the vast improvement in AP, nearly 27.9% for Fin whales. These results indicate that the optimal configuration for the



(a) 8,192 Sample Window $64 \times 256 \times 1$ Feature Size

(b) 8,192 Sample Window $128 \times 128 \times 1$ Feature Size

(c) 8,192 Sample Window $256 \times 64 \times 1$ Feature Size

(d) 16,384 Sample Window $64 \times 512 \times 1$ Feature Size

(e) 16,384 Sample Window $128 \times 256 \times 1$ Feature Size

(f) 16,384 Sample Window $256 \times 128 \times 1$ Feature Size

(g) 32,768 Sample Window $128 \times 512 \times 1$ Feature Size

(h) 32,768 Sample Window $256 \times 256 \times 1$ Feature Size

(i) 32,768 Sample Window $512 \times 128 \times 1$ Feature Size

Fig. 7: The proposed model's Precision-Recall curves trained with different window and feature sizes. The window size is the number of audio vector samples used as input for the model, and the feature size is the size of the spectrogram extracted from each vector.

Fin Whale task is the previously mentioned parameters, trained as a multi-label problem with blue whale as the second class.

Figure 5, 6, and 7 provides a comprehensive view of the training done by all models in the ablation study. In most cases, the validation accuracy incrementally increases while the loss also tends to increase. These observations could be explained by the model trying to resolve discrepancies between annotations and the data. Moreover, a window size of 16,384 and a feature size of $128x256$ has the

best PR curve regarding Fin Whales while retaining similar performance for Blue Whales.

6. Discussion

This section presents and discusses the study's main findings, limitations, and implications. We also suggest some directions for future work for possible improvements upon our method.

1. **Large Vs Small:** One of the key features of our model is the use of larger temporal windows, which allows us to capture more whale calls in more spectrograms. As such, our model differs from previous works that used smaller windows of 4.5 or 9 seconds, such as Miller et al. [9] and Rasmussen et al. [8]. Given that the primary intuition of using a small window was to isolate the average 3-second whale D-Call within the window, these techniques do not apply to the multi-label classification problem due to the variability in the duration of different Blue Whale calls [3,7]. Also, short-duration windowing severely limits the feature size of the STFTs generated for CNN classification. Because this factor does not limit us, our method can use some larger, more complex model designs across an array of STFT parameter choices, allowing improvement in frequency and temporal resolution balance.
2. **Dataset Curation:** The chunking methods address truncated positive features. A smaller window exacerbates these effects by creating more fragmented features, thereby increasing the need to curate the dataset by discarding partial positive samples. Our large temporal window mitigates the number of truncated features, ensuring that the method captures at least one complete sample of a positive annotation for any annotation in the dataset. This will ensure that our proposed model can help capture the expansive spectral and temporal features of whales from the entire positive set of samples, instead of just a subset.
3. **Dataset Imbalance:** A smaller window significantly increases the number of negative samples, necessitating substantial over or under-sampling to resolve the issue. Under-sampling the negative class, while computationally feasible, results in the exclusion of a substantial data pool that may be vital for comprehensive model training. Conversely, our approach with a larger window size addresses these concerns by bolstering the count of positively annotated samples by reducing the necessity to discard partial positives. Moreover, the larger window size diminishes the total number of negative samples within the dataset, adding a second correcting factor for dataset imbalance. For instance, in the methodology proposed by Miller et al. [9], a mere $77,000$ out of over 3 million negative samples were utilized for training. This approach induces a bias, as the model is trained on a relatively diverse set of positive samples but a limited set of negatives. Consequently, training on a dataset that does not accurately reflect the true class distribution necessitates extensive threshold adjustments to attain balanced outcomes. On the other hand, our model significantly decreases

dataset imbalances, thus reducing the need to perform dataset imbalance resolution techniques and keeping the training data similar to the true data distribution. This is reflected in that we did not need to perform threshold tuning to meet or exceed existing methods.

4. **Augmentation Layers:** Another feature of our model is the use of additional training augmentation layers, such as Gaussian noise, horizontal flipping, and random translation, which aim to improve the robustness and generalization of our model. These layers helped to reduce the number of false positives and increase the recall scores on the datasets. However, our study did not conduct an ablation study to evaluate the specific effect of each layer and its impact on the model under a range of parameters.
5. **Threshold Optimization:** The strategic adjustment of decision thresholds in DeepWhaleNet can significantly bolster precision while maintaining an acceptable level of recall. By calibrating the threshold, the model is fine-tuned to favor highly confident predictions, which is crucial for the accurate classification of infrequent whale species like Blue and Fin Whales, where the cost of false positives outweighs that of false negatives. Although a higher threshold may lead to a modest decrease in recall, models with an elevated Average Precision (AP) demonstrate resilience against such declines. We advocate for a methodical threshold optimization process, beginning with the establishment of the decision threshold on a validation set, followed by evaluation on a separate test set. This approach circumvents the potential bias that could arise from direct threshold determination on test data. Future research should delve into the synergistic effects of various augmentation techniques, loss functions, and threshold settings to further refine classification efficacy.
6. **Loss of Locality:** One drawback to the larger window is the loss of locality for whale calls in each sample, which helps identify the exact position and duration of the whale call in the audio file. To address this limitation, we propose an interesting technique that uses salience mapping, which computer vision and other domains widely use. Salience mapping is a method of generating a map that highlights the regions of the input that are most relevant or influential for the output [24]. By applying salience mapping to our model, reverse correlating the salience map with the signal samples can retrieve the vocalization event's exact location. This technique could not only restore the locality of the whale call but also provide visual feedback and confidence to the users of our model.
7. **Pretrained Architectures:** Based on experimentation, utilization of pretrained architectures is not advisable. A large concern is computational efficiency. The augmentation of feature dimensions, coupled with the tripling of a single-channel matrix, results in the new feature occupying approximately 4.59 times more system memory space. This increment is particularly detrimental in scenarios where computational resources are limited by memory capacity, as a pretrained architecture would significantly amplify this limitation. Addition-

ally, the computational speed is markedly impacted. The process of feature generation necessitates resizing, which further complicates the feature creation workflow. For instance, processing $35,000$ spectrograms with the resized triple-channel matrix requires 322 seconds, whereas the non-resized variant completes in merely 32 seconds, translating to a speed-up of over 10 times. Similar performance is observed in predictive and training operations. Utilizing an Ampere 80GB GPU, a single training step with a batch size of 64 is executed in 114 ms with the pretrained architecture. In contrast, our streamlined custom architecture accomplishes the same step in 32 ms, yielding a 3.56-fold acceleration. DeepWhaleNet not only surpasses the pretrained architecture in performance but also demonstrates superior efficiency across various processing stages.

The results of our study underscore the efficacy and adaptability of the proposed DeepWhaleNet model across various tasks and configurations. A key observation is the trade-off between precision and recall, which necessitates careful consideration based on the specific requirements of the task at hand. Above, we suggest future work for exploring strategies to enhance this balance, such as adjusting the classification threshold or employing different loss functions. Additionally, further experiments could delve into the impact of various factors on the model's performance, including the type of augmentation layers, learning rate, data cleaning, or employing other machine learning methods like active learning [25].

7. Conclusion and Future Works

In this study, we have detailed the effects of varying STFT parameters on audio signal analysis. Our results highlight the delicate balance required for optimal parameter selection, which is crucial for accurate audio classification. As we present our concluding thoughts, we reflect on the significance of these findings for the field of audio signal processing.

Our study has introduced various windowing methodologies tailored for Machine Learning algorithms in the context of Underwater Passive Acoustic Monitoring data. Our proposed framework, termed DeepWhaleNet, enhances the benchmark performance and exhibits versatility across call-specific, species-specific, and multi-label detection tasks. Our paper demonstrated that employing larger window sizes affords the model an expanded repertoire of STFT parameters, thereby facilitating a broader range of frequency and temporal resolutions suitable for more intricate models. Furthermore, DeepWhaleNet effectively mitigated dataset imbalances by reducing the volume of negative samples and enriching the diversity of positive samples. Importantly, the incorporation of augmentation layers within the model architecture significantly bolstered the robustness of our whale detection system; not to mention, we offer strategies to address the minor limitations associated with the adoption of larger windowing techniques.

The tools developed through this research could revolutionize the practices of conservationists, environmental managers, and marine biologists by improv-

ing their abilities to survey and monitor whale distributions. Such advancements are instrumental in the accumulation of longitudinal data on the dynamics of the whale population, thereby informing dynamic and adaptive management strategies [26]. An intriguing prospect lies in the potential synergy between structured wildlife research, artificial intelligence, and citizen science, mainly through partnerships with non-profit organizations like Tech4Wildlife, to propel the frontiers of conservation science.

Looking ahead, our research agenda includes the integration of active learning paradigms and other cutting-edge methodologies to refine the reliability of dataset annotations in the absence of human oversight. Also, a key objective is to restore the spatial specificity of whale vocalizations within spectrograms. Moreover, our objective was to assess the detection efficacy of the model across various demographics of whales, depending on the availability of pertinent datasets. Lastly, exploring more sophisticated computer vision architectures is anticipated to elevate the precision and recall metrics of whale identification further.

Code

To facilitate the reproduction of our experiments, we will make public a GitHub repository with our proposed model upon publication at: https://github.com/2ai-lab/DeepWhaleNet

Acknowledgments

Computational Resources were provided by the Lawrence High Performance Computing Supercomputer. Funded by NSF Grants: 1626516 and 2346643.

References

1. Joe Roman, James A Estes, Lyne Morissette, Craig Smith, Daniel Costa, James McCarthy, J Brian Nation, Stephen Nicol, Andrew Pershing, and Victor Smetacek. Whales as marine ecosystem engineers. *Frontiers in Ecology and the Environment*, 12(7):377–385, 2014.
2. Trish J Lavery, Ben Roudnew, Peter Gill, Justin Seymour, Laurent Seuront, Graham Johnson, James G Mitchell, and Victor Smetacek. Iron defecation by sperm whales stimulates carbon export in the southern ocean. *Proceedings of the Royal Society B: Biological Sciences*, 277(1699):3527–3531, 2010.
3. Brian S. Miller, Brian S. Miller, Kathleen M. Stafford, Ilse Van Opzeeland, Danielle Harris, Flore Samaran, Ana Širović, Susannah Buchan, Ken Findlay, Naysa Balcazar, and et al. An open access dataset for developing automated detectors of antarctic baleen whale sounds and performance evaluation of two commonly used detectors. *Scientific Reports*, 11(1), 2021.
4. Devis Tuia, Benjamin Kellenberger, Sara Beery, Blair R Costelloe, Silvia Zuffi, Benjamin Risse, Alexander Mathis, Mackenzie W Mathis, Frank van Langevelde, Tilo Burghardt, et al. Seeing biodiversity: perspectives in machine learning for wildlife conservation. *Nature Communications*, 12(1):1–13, 2021.
5. Elena Schall and Clea Parcerisas. A robust method to automatically detect fin whale

acoustic presence in large and diverse passive acoustic datasets. *Journal of Marine Science and Engineering*, 10(12):1831, 2022.
6. Ana Širović. Variability in the performance of the spectrogram correlation detector for north-east pacific blue whale calls. *Bioacoustics*, 25(2):145–160, 2015.
7. Shannon Rankin, Don Ljungblad, Chris Clark, and Hidehiro Kato. Vocalisations of antarctic blue whales, balaenoptera musculus intermedia, recorded during the 2001/2002 and 2002/2003 iwc/sower circumpolar cruises, area v, antarctica. *J. Cetacean Res. Manage.*, 7(1):13–20, 2023.
8. Jeppe Have Rasmussen and Ana Širović. Automatic detection and classification of baleen whale social calls using convolutional neural networks. *The Journal of the Acoustical Society of America*, 149(5):3635–3644, 2021.
9. Brian S. Miller, Shyam Madhusudhana, Meghan G. Aulich, and Nat Kelly. Deep learning algorithm outperforms experienced human observer at detection of blue whale d-calls: A double-observer analysis. *Remote Sensing in Ecology and Conservation*, 9(1):104–116, 2022.
10. Mark Baumgartner. Guide to monitoring real-time marine mammal detections using autonomous platforms. Technical report, Woods Hole Oceanographic Institution, 2016.
11. Charles Scott Baker, Diane Claridge, Charlotte Dunn, Thomas Fetherston, Dorothy Nevé Baker, Holger Klinck, and Debbie Steel. Quantification by droplet digital pcr and species identification by metabarcoding of environmental (e)dna from blainville's beaked whales, with assisted localization from an acoustic array. *PLOS ONE*, 2023.
12. James A Theriault, Harald Yurk, and Hilary B Moors-Murphy. Workshop report: review of near-real time whale detection technologies. Technical Report 3410, Fisheries and Oceans Canada, 200 Kent Street, Ottawa, Ontario, K1A 0E6, 2020.
13. Ari Friedlaender, Michelle Modest, and Chris Johnson. Whales of the antarctic peninsula, 2020. Contributors: David Johnston, Jennifer Jackson, Sarah Davie. Acknowledgements: Rod Downie, Reinier Hille Ris Lambers, Rick Leck, Duke University Marine Robotics and Remote Sensing Lab, California Ocean Alliance, One Ocean Expeditions.
14. Flore Samaran, Kathleen M Stafford, Trevor A Branch, Jason Gedamke, Jean-Yves Royer, Robert P Dziak, and Christophe Guinet. Seasonal and geographic variation of southern blue whale subspecies in the indian ocean. *PLOS ONE*, 2013.
15. Ana Širović, Lauren N Williams, Sara M Kerosky, Sean M Wiggins, and John A Hildebrand. Temporal separation of two fin whale call types across the eastern north pacific. *Marine Biology*, 160:47–57, 2013.
16. Russel D. Andrews Richard Sears Véronique Lesage, Katherine Gavrilchuk. Foraging areas, migratory movements and winter destinations of blue whales from the western north atlantic. *Endangered Species Research*, 34:27–43, 2017.
17. Andrew McCallum, Chris Nugent, Ian Cleland, and Paul McCullagh. A comparative analysis of windowing approaches in dense sensing environments. *Proceedings*, 2(19):1245, 2018.
18. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Dec 2015.
19. Duling Xv; Li Yang. Research on urban audio classification based on residual neural network. In *2021 International Conference on Computer Engineering and Application (ICCEA)*. IEEE, 2021.
20. Laith Alzubaidi et al. Deepening into the suitability of using pre-trained models of imagenet against a lightweight convolutional neural network in medical imaging: an experimental study. *PeerJ Computer Science*, 7:e715, 2021.
21. Artem Guzhov, Fabian Raue, Jörn Hees, and Andreas Dengel. Comparison of pre-

trained cnns for audio classification using transfer learning. *Sensors*, 10(4):72, 2021.
22. Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference for Learning Representations*, 2015.
23. Shiv Ram Dubey, Satish Kumar Singh, and Bidyut Baran Chaudhuri. Activation functions in deep learning: A comprehensive survey and benchmark. *Neurocomputing*, 503:92–108, Sep 2022.
24. Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, page 2921–2929, 2016.
25. Mohamed-Rafik Bouguelia, Slawomir Nowaczyk, K. C. Santosh, and Antanas Verikas. Agreeing to disagree: Active learning with noisy labels without crowdsourcing. *International Journal of Machine Learning and Cybernetics*, 9(8):1307–1319, 2017.
26. Caleb Scoville, Melissa Chapman, Razvan Amironesei, and Carl Boettiger. Algorithmic conservation in a changing climate. *Current Opinion in Environmental Sustainability*, 51:30–35, Aug 2021.