Robust Detection of North Atlantic Right Whales using Deep Learning Methods

William Alexander Vickers

School of Computing Sciences

University of East Anglia

This thesis is submitted for the degree of

Doctor of Philosophy

December 2021

Acknowledgements

Firstly, I would like to thank Dr Ben Milner, whom without his supervision, drive, patience and academic expertise, my work would not be where it is today. Throughout the last four years we have developed a close working relationship and become good friends, for which I will miss our twice weekly meetings. Your continuous support and belief in me has genuinely pushed me to do something I never thought possible and for that I am extremely grateful.

I would also like to thank Dr Jason Lines for his supervision and willingness to help whenever I needed it. I appreciate all the advice and knowledge you have passed on. I would like to sent my appreciation to both Dr Denise Risch and Rob Lee for their supervision and input throughout this project.

Much of this work would not be possible without the motivation and help I have received from the friends I have met and made along the way, namely Michael Flynn, James Large, Nick Matthews, Josh Thody and everybody in the Machine Learning and Speech labs, along with everybody else in the School of Computing Sciences at UEA that have added to my experience. To my friends outside of the university, you have all kept me stimulated, academically curious over the last four years, provided the richest conversations and have always been interested in learning about this work, for that I thank you all. I would like to say a special thanks to Jack Pettitt for the

endless phone conversations and words of wisdom and as he once said - *"lots of small hills make a mountain"*.

A special thanks also go to my parents, and sister for their belief in me and continued support in all parts of my life, which has enabled the production of this thesis.

Finally, I would like to thank my partner, Tabitha Reuben who has contributed positively in every possible way to my development and success as a PhD student. She has provided endless support and sacrifice in making this research possible and without her I would not have achieved what I have.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this thesis are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This thesis is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

William Alexander Vickers

December 2021

Publications

As first author

- Vickers, W., Milner, B., Risch, D., & Lee, R. (2021). Robust North Atlantic right whale detection using deep learning models for denoising. The Journal of the Acoustical Society of America (JASA), 149(6), 3797-3812.

- Vickers, W., Milner, B., & Lee, R. (2021, May). Improving The Robustness Of Right Whale Detection In Noisy Conditions Using Denoising Autoencoders And Augmented Training. In 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 91-95).

- Vickers, W., Milner, B., Gorpincenko, A., & Lee, R. (2020, September). Methods to Improve the Robustness of Right Whale Detection using CNNs in Changing Conditions. In 2020 28th European Signal Processing Conference (EUSIPCO) (pp. 106-110). IEEE.

- Vickers, W., Milner, B., Lee, R., & Lines, J. (2019, September). A comparison of machine learning methods for detecting right whales from autonomous surface vehicles. In 2019 27th European Signal Processing Conference (EUSIPCO) (pp. 1-5). IEEE.

- Vickers, W., Milner, B., Lines, J., & Lee, R. (2019, September). Detecting right whales from autonomous surface vehicles using RNNs and CNNs. In 2019 27th European Signal Processing Conference Workshop (EUSIPCO).

- Vickers, W., Milner, B., & Lee, R. (2019, May). Reducing processing requirements for right whale detection from autonomous surface vehicles. In 2019 Proceedings of the Institute of Acoustics (IOA).

As contributing author

- Middlehurst, M., Vickers, W., & Bagnall, A. (2019, November). Scalable dictionary classifiers for time series classification. In International Conference on Intelligent Data Engineering and Automated Learning (pp. 11-19). Springer, Cham.

Abstract

This thesis begins by assessing the current state of marine mammal detection, specifically investigating currently used detection platforms and approaches of detection. The recent development of autonomous platforms provides a necessity for automated processing of hydrophone recordings and suitable methods to detect marine mammals from their acoustic vocalisations. Although passive acoustic monitoring is not a novel topic, the detection of marine mammals from their vocalisations using machine learning is still in its infancy. Specifically, detection of the highly endangered North Atlantic right whale (*Eubalaena glacialis*) is investigated. A large variety of machine learning algorithms are developed and applied to the detection of North Atlantic right whale (NARW) vocalisations with a comparison of methods presented to discover which provides the highest detection accuracy. Convolutional neural networks are found to outperform other machine learning methods and provide the highest detection accuracy when given spectrograms of acoustic recordings for detection.

Next, tests investigate the use of both audio and image based enhancements method for improving detection accuracy in noisy conditions. Log spectrogram features and log histogram equalisation features both achieve comparable detection accuracy when tested in clean (noise-free), and noisy conditions.

Further work provides an investigation into deep learning denoising approaches, applying both denoising autoencoders and denoising convolutional neural networks to noisy NARW vocalisations. After initial parameter and architecture testing, a full evaluation of tests is presented to compare the denoising autoencoder and denoising convolutional neural network. Additional tests also provide a range of simulated real-world noise conditions with a variety of signal-to-noise ratios (SNRs) for evaluating denoising performance in multiple scenarios. Analysis of results found the denoising autoencoder (DAE) to outperform other methods and had increased accuracy in all conditions when testing on an underlying classifier that has been retrained on the vestigial denoised signal. Tests to evaluate the benefit of augmenting training data were carried out and discovered that augmenting training data for both the denoising autoencoder and convolutional neural network, improved performance and increased detection accuracy for a range of noise types.

Furthermore, evaluation using a naturally noisy condition saw an increase in detection accuracy when using a denoising autoencoder, with augmented training and convolutional neural network classifier. This configuration was also timed and deemed capable of running multiple times faster than real-time and likely suitable for deployment on-board an autonomous system.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Motivation

Cetacea is the phylogenetic infraorder that contains all species of cetaceans, otherwise, known as whales. North Atlantic right whales (NARWs) from the baleen family are currently among the most endangered cetaceans, with fewer than 400 remaining [61]. Although North Atlantic right whales pose no threat to human life and feed on zooplankton [61], their population has declined since 2010 after an initial reduction in numbers prior to 1935 when the United States outlawed whaling [61, 78]. A recent report [147] has found deaths of NARWs since 2010 to be predominately caused by entanglements with fishing equipment, and a small percentage due to ship strikes. Since the human population is increasing year on year [57], there has been a noticeable increase in food production [47], leading to increased fishing efforts [21]. Delivery and movement of goods worldwide have also seen huge increases, and as such, shipping traffic has soared, causing a severe impact on the population of NARWs and other species [159]. With the number of breeding female NARWs estimated at less than 100 [61], it is essential that protection and mitigation efforts are put in place to ensure the long-term survival of the species.

1.2 Methods of North Atlantic right whale detection

Observations via ships [23] and aircraft [60], or acoustic recordings to listen for vocalisations, are all examples of traditional techniques to monitor NARWs. With the development of smaller and more affordable computer components, monitoring efforts have shifted to a more automated approach with human observers analysing recordings after collection from buoys or autonomous vehicles [15, 192, 40]. In addition, a shift to computerised data collection has highlighted the requirement for automated data processing, aiming to match the performance of a human observer. This work explores techniques to automatically process recordings and provide accurate feedback, much like that given by a human expert. The automated system aims to process data in real-time so feedback can be instant and ensures a backlog of unprocessed data is not created.

Techniques for detecting whales have varied dramatically in recent years, with a traditional system focused on monitoring amplitude rises within audio [229, 70] to indicate the presence of a whale. More advanced systems have aimed to track frequency contours within the spectral-domain of a signal [154] or use pre-existing speech recognition techniques such as hidden Markov models (HMMs) [7] or Gaussian mixture models (GMMs) [168, 235]. The early 2010s has seen the rise of neural network based classifiers for image classification [106, 82], with expansion into many other domains gaining traction [76].

1.3 Aims and objectives

In order to take advantage of these techniques, this work aims to apply neural network approaches, amongst more traditional machine learning classifiers, to provide the best solution for detecting NARWs. A gain in performance that neural network classifiers have shown in other areas could be vital in creating a robust NARW detection system for use in real-world conditions.

Although the problem of acoustically recognising NARWs is both a detection and classification problem, this work focuses on classifying NARW vocalisations from other background sources, either natural or anthropogenic. The problem of acoustic *detection* more closely investigates how a constant stream of audio is processed in order to alert a user of a noise presence. The problem of acoustic *classification* therefore aims to distinguish specific classes from the detected audio. The aim of the system built and developed throughout this work is to process segments of audio to make a classification of whether a NARW vocalisation is present. As this work is primarily aimed at developing and refining the proposed classification system it is standalone for use within a wider range of use cases. For example it could be adapted for marine mammal collision avoidance, population monitoring, or event classification, however exploring further implementation of the classification system falls outside the boundary of this work and should be explored in future projects.

1.4 Structure of thesis

The remainder of this thesis will be structured as followed.

Chapter 2: This chapter provides a background for the application of NARW classification. First, the Cetacea infraorder is explored, analysing each suborder and its vocalisation characteristics. The motivation behind the project then follows, with a review of current detection platforms. Finally, the data used throughout this project is detailed.

Chapter 3: In this chapter, a range of baseline machine learning methods are applied to the problem of NARW detection. Applied algorithms are split into two groups, time series and deep learning. Time series methods are used as a benchmark to compare against the more exploratory deep learning methods; the performance from these algorithms are well documented, however, the most suitable configurations of deep learning architectures are still being researched. A range of deep learning approaches are reviewed with experimental architectures considered for each method. Lastly, a comparison of the top performing algorithms are presented, discussing the best overall method for NARW detection.

Chapter 4: This chapter begins by focusing on the effect of noise within acoustic recordings and the effect this can have on the performance of a detection system. Chapter 4 tests a range of conventional image and audio enhancement methods, to explore their impact on reducing noise within recordings. The suitability of each noise reduction technique is assessed by monitoring the performance of the detection system, which mimics real-world use, expecting detection accuracy to increase with a higher reduction of noise. Finally, audio enhancement methods are compared against image enhancement methods applied to spectrograms to investigate whether noise reduction should occur before conversion to the spectral domain, or not.

Chapter 5: This chapter builds on work from Chapter 4 and explores the use of autoencoders for noise reduction. Autoencoders are a neural network architecture which provides self-feedback between input features, output features and target features. First, autoencoders are trained to output clean (non-noisy) spectrograms from noisy spectrograms. Further developments use denoising autoencoders (DAE) which utilise noisy-clean pairs, and aim to learn the difference between the clean and noisy and produce denoised spectrograms for more accurate classification.

Chapter 6: This chapter introduces a noise reduction technique called denoising convolutional neural network (DNCNN). This algorithm uses a similar approach to the autoencoder, utilising convolutional layers to encode the input features. Instead of using a specific compression architecture like the autoencoder, the DNCNN aims to predict noise within the noisy segment using a convolutional neural network architecture; noise can then be subtracted from the noisy segment with the aim of producing a denoised spectrogram.

Chapter 7: This chapter investigates both the DAE from Chapter 5 and DNCNN from Chapter 6 across a range of noise corruptions and signal-to-noise ratios. Both methods are also investigated with the use of augmented training to discover the optimal denoising and classification system for NARW vocalisations in noisy conditions. Two methods of classifier training are also evaluated. Using clean data and using vestigial data that has been previously denoised and likely matches the test data more closely.

Chapter 8: An unsupervised method of domain adaptation is investigated in this Chapter, to assess performance in changing conditions when augmentation of a current model or training a new model is infeasible.

Chapter 9: This thesis concludes with a summary of the experiments carried out, what has been learnt from these tests, and what it means for current NARW detection. Finally, this chapter ends with a discussion of future work and how this work can be built upon in subsequent research.

Chapter 2

Background

2.1 Introduction

This chapter provides the background and motivation for the technical work presented within this thesis. Initially, the Cetacea infraorder within the phylogenetic tree is explored. This lays out the biological order of cetaceans, which encompasses two suborders: Mysticeti and Odontoceti. Finally, traversing the Cetacea order provides insights into similarities between species and influences suitable detection methods to later explore.

In Section 2.3 the motivation for the project is detailed. Specifically, this examines the need for an autonomous method of marine mammal detection and why acoustic methods are the most suitable and sustainable. Anthropogenic factors, such as fishing entanglement and vessel strikes [61], currently threaten marine mammal populations to critically low levels [147]. Mitigation measures try to reduce, avoid and offset the adverse effects of human interactions to improve population numbers. As populations of many marine mammals continue to fall [141], a constant review of current mitigation methods is essential to ensure protection efforts are optimal.

A review of detection platforms is discussed next; focusing on currently used platforms, what form future platforms could take, how they would work and why they could be more suitable than the current solutions. As advancements in autonomous technologies emerge and computing hardware becomes smaller, cheaper and more power efficient, it is important to revise previous solutions in order to gain the benefits from new technologies.

To set out the data used within this thesis, Section 2.6 details the NARW vocalisation data used, where the data has been collected and how it has been used or manipulated for later experiments. Explicitly detailing information about the data enables tests to be repeatable whilst also making the difference between each corpus clear when trying understand variation in test results. In order to generate an in-depth analysis and provide more insightful experimental results, much of the data used has been augmented with noise. Noise augmentation can be carried out in a number of ways and Section 2.6.4 explains the process taken to ensure noise is added appropriately, reflecting real world environmental noise conditions.

Finally, Section 2.7 reviews techniques that have been traditionally used to survey areas of ocean and detect marine mammals. Traditional detection methods rely less on modern machine learning approaches and make use of fundamental signal processing practices to detect cetaceans.

2.2 Cetacea phylogenetics

The phylogenetic tree is a tree diagram that depicts the evolutionary descent lines for different species [13]. Phylogenies help map the structure and groupings of specific species. The scope of this thesis is exclusively concerned with the Cetacea order. Cetacea defines the order at which all marine mammals are collected and mapped. Within this order, suborders exist to categorise cetaceans into two additional feature defining groups: Mysticeti and Odontoceti [63]. Mysticeti cetaceans are also known as baleen whales and Odontoceti, known as toothed whales [63]. The main difference between the suborders is their feeding mechanism, with baleen whales using baleen plates to filter food whilst toothed whales have teeth [131].

Baleen whales are some of the largest inhabitants of the modern ocean, weighing up to 190 tonnes and ranging in length from 6.5m to 33m in the largest mammals [131]. The number of baleen whale species is dramatically less than toothed whales, with 16 species in existence compared to 76 species of toothed whales. Baleen whales lack teeth and instead use a comb-like structure called baleen. The water inside the mouth is pushed through the baleen, filtering out vast amounts of prey [131] during feeding. Figure 2.1 depicts the size of many cetaceans and puts the size of baleen whales into perspective compared to humans and other land mammals. Baleen whales, unlike toothed whales, do not use echolocation for hunting [227]. The production mechanism for how cetaceans produce sound is highly debated; however, vocal folds are present within the animal's larynx and are commonly thought to be responsible for sound production [227]. All baleen whales produce low-frequency sounds (<10kHz) compared to the highest frequency produced by toothed whales that can reach 130kHz. In general, however, excluding minke whales [19] and some singing whales [33], baleen whales use

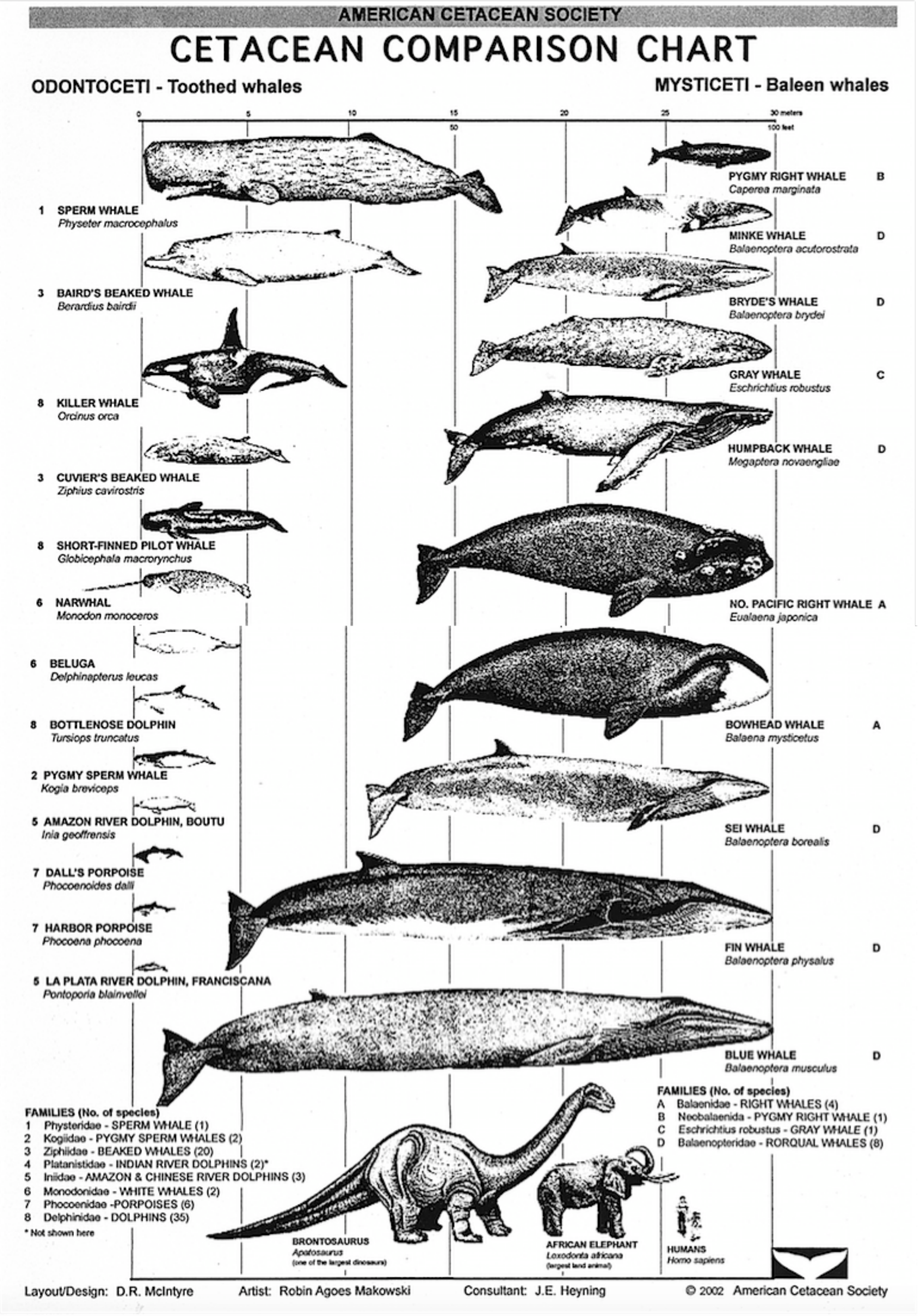

AMERICAN CETACEAN SOCIETY
CETACEAN COMPARISON CHART
ODONTOCETI - Toothed whales
MYSTICETI - Baleen whales
0
5
10
15
20
25
30 meters
50
100 feet
1 SPERM WHALE
Physeter macrocephalus
3 BAIRD'S BEAKED WHALE
Berardius bairdii
8 KILLER WHALE
Orcinus orca
3 CUVIER'S BEAKED WHALE
Ziphius cavirostris
8 SHORT-FINNED PILOT WHALE
Globicephala macrorynchus
6 NARWHAL
Monodon monoceros
6 BELUGA
Delphinapterus leucas
8 BOTTLENOSE DOLPHIN
Tursiops truncatus
2 PYGMY SPERM WHALE
Kogia breviceps
5 AMAZON RIVER DOLPHIN, BOUTU
Inia geoffrensis
7 DALL'S PORPOISE
Phocoenoides dalli
7 HARBOR PORPOISE
Phocoena phocoena
5 LA PLATA RIVER DOLPHIN, FRANCISCANA
Pontoporia blainvellei
PYGMY RIGHT WHALE B
Caperea marginata
MINKE WHALE D
Balaenoptera acutorostrata
BRYDE'S WHALE D
Balaenoptera brydei
GRAY WHALE C
Eschrichtius robustus
HUMPBACK WHALE D
Megaptera novaengliae
NO. PACIFIC RIGHT WHALE A
Eualaena japonica
BOWHEAD WHALE A
Balaena mysticetus
SEI WHALE D
Balaenoptera borealis
FIN WHALE D
Balaenoptera physalus
BLUE WHALE D
Balaenoptera musculus
FAMILIES (No. of species)
1 Physteridae - SPERM WHALE (1)
2 Kogiidae - PYGMY SPERM WHALES (2)
3 Ziphiidae - BEAKED WHALES (20)
4 Platanistidae - INDIAN RIVER DOLPHINS (2)*
5 Iniidae - AMAZON & CHINESE RIVER DOLPHINS (3)
6 Monodonidae - WHITE WHALES (2)
7 Phocoenidae -PORPOISES (6)
8 Delphinidae - DOLPHINS (35)
* Not shown here
FAMILIES (No. of species)
A Balaenidae - RIGHT WHALES (4)
B Neobalaenida - PYGMY RIGHT WHALE (1)
C Eschrichtius robustus - GRAY WHALE (1)
D Balaenopteridae - RORQUAL WHALES (8)
BRONTOSAURUS
Apatosaurus
(one of the largest dinosaurs)
AFRICAN ELEPHANT
Loxodonta africana
(largest land animal)
HUMANS
Homo sapiens
Layout/Design: D.R. McIntyre
Artist: Robin Agoes Makowski
Consultant: J.E. Heyning
© 2002 American Cetacean Society

Fig. 2.1 A cetacean size comparison chart created by the American Cetacean Society to represent the scale of cetaceans [127]. For comparison a human, elephant and brontosaurus are included. The chart is divided into Odontocetes (left) and Mysticetes (right).

frequencies in the 5Hz - 1kHz range.

Toothed whales are a much larger suborder, with 76 species currently in existence. All dolphins, porpoises and beaked whales are included within this suborder, making it more diverse than the baleen whale group; however they are much smaller in physical size ranging from 1.5m to 20m [228]. One of the key differentiators between these Cetacea suborders is the large ovoid melon in the anterior part of the facial region that toothed whales possess [134]. This fatty tissue is suspected to be a vital component of the echolocation system of which the baleen whale equivalent is dramatically smaller [228]. Toothed whales are thought to have developed specialised sound production and reception mechanisms for handling sonar signals much like those found on bats [228]. These mechanisms are responsible for producing clicks, pulses and whistles. Clicks are known to be used for echolocation [98], whilst pulses and whistles for communication [228]. Furthermore, sound production from toothed whales are significantly more varied than baleen whales, with frequencies up to 130kHz for some species of dolphin.

Understanding the differences between cetacean species is crucial for evaluating suitable detection methods. Since certain baleen whale calls operate within a similar frequency range, it indicates that a singular method to detect their vocalisations could be successfully applied to multiple species without further manipulation, or species specific parameterisation.

2.3 Motivation

The North Atlantic right whale (NARW) is the main focus of this thesis, with algorithmic development and experimental results reflecting this. The decision to focus on NARWs is made up of two factors; first, in 2020, NARWs were one of the most critically endangered marine mammals [147]; second, due to their extinction risk they have been extensively investigated - subsequently, well documented, accurate and structured data-sets exist of their vocalisations, providing a strong starting point for acoustic experimental work.

The North Atlantic right whale has the scientific name *Eubalaena glacialis*, and is part of the Balaenidae family and is 1 of 16 current baleen whales. NARWs are among the largest marine mammals growing up to 15.5m in length and weighing up to 65 tonnes [61]. They generally have a lifespan up to 70 years; however, human activity can drastically reduce this life expectancy [61]. As previously mentioned, NARWs are currently critically endangered, with an estimated 360 remaining and have a declining population [147]. The decline of the NARW population was largely due to commercial whaling bringing the species close to extinction before being declared illegal in 1935 by the U.S. government [61]. In 1992, it was estimated that only 295 were alive; however, numbers increased year on year until 2010 when the population began to decline again [61], this trend can be seen in Figure 2.2. Of recent NARW deaths, humans are thought have caused all of them, mainly through fishing entanglement and vessel strikes [61]. In contrast, the population of southern right whales (SRWs) has increased in recent years [10]. It is hypothesised that their habitat contains far less human activity and therefore chance of death.

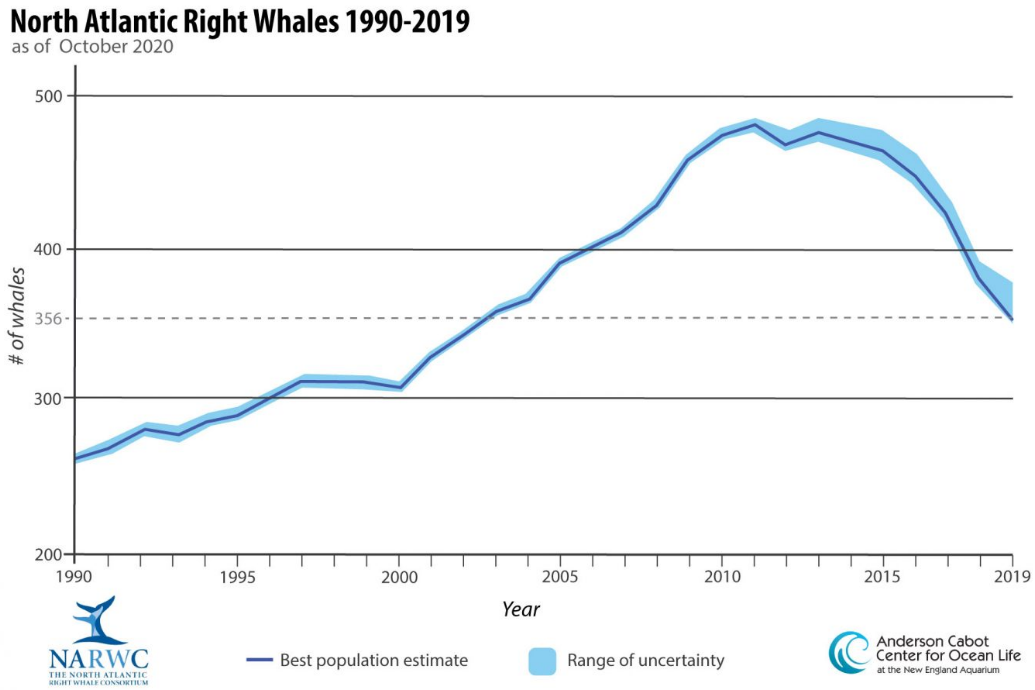


Fig. 2.2 A chart created by the Anderson Cabot Center for Ocean Life [78] detailing the North Atlantic right whale population trend from 1990-2019.

2.3.1 Extinction

North Atlantic right whales are protected under both the Endangered Species Act (ESA) [194] and Marine Mammal Protection Act (MMPA) [48] since the current population count is critically low [62]. Since the North Atlantic ocean is also shared with Canada, the Canadian government have also recently introduced measures with the aim of preventing harm to NARWs. One of these is an interim order to protect NARWs [27]. Although the NARW has been on the ESA since 1970 [62], numbers have further declined since 2010, emphasising that protection is necessary to save them from extinction. Due to their recent decline in numbers it is more important than ever to develop reliable methods of detection that can be used in harsh environmental conditions, in all hours of the day, where humans cannot operate.

To fight extinction of any species, conservation efforts must be bought into place in order to help the species prosper and breed healthy numbers of offspring. Since NARWs have long lifespans, ensuring populations rise is centered around keeping those remaining, alive and healthy. A report by the Woods Hole Oceanographic Institution (WHOI) collated research surrounding the deaths of NARWs from 1970-2015, finding that 85% of deaths from 2010-2015 were due to entanglement with fishing ropes, with the remaining 15% caused by vessel strikes [147]. As 100% of deaths from 2010-2015 were due to anthropogenic activities, reducing needless deaths is solely down to changing human behaviour and its effect on the oceanic environment. This can either be done by stopping all human-lead oceanic events such as shipping, fishing, and construction, or by monitoring the location of NARWs to mitigate the effect that human activities previously had on them.

Although serious mitigation techniques may be necessary, current mitigation measures are less dramatic than complete operation shutdowns and include techniques such a reduced speed for shipping lanes, closures at specific times and designated shipping lanes that must be adhered too.

In order to decrease the number of NARW deaths, the National Oceanic Atmospheric Administration (NOAA) is responsible for planning and implementing recovery of the NARW population, *"with an interim goal of down-listing its status from endangered to threatened"* [62]. Within the NOAAs plan for recovery of the species, key points include [62]:

- protection of habitats to ensure breeding safety
- minimal effect of vessel activity
- monitoring of population size and movement trends

This recovery plan relies heavily on monitoring populations and accurately detecting the presence of a NARW in any given location. Being able to accurately detect NARWs is therefore paramount in the fight to stop their extinction.

2.3.2 Monitoring

NARWs are largely coastal marine mammals but are also known to travel in open oceans for long periods [61]. The Atlantic ocean is the second largest body of water on earth and covers 20% of the earths surface [128]. Due to the expanse of the Atlantic ocean, sighting and counting marine species is an extremely challenging task that requires expert knowledge, expensive and specialised equipment, and is an impractical task for regularly monitoring the location and population of marine mammals. With current detection methods being outdated by the technology they use, it is vital that new techniques are developed to reap the benefits of new technologies and to keep up with the growing need to monitor the oceans. As previously mentioned, the recovery plan set out by the NOAA [62] focuses on being able to accurately detect a NARW in any given location and provide feedback to enable mitigation techniques to be actionable. This NOAA recovery plan makes it clear that it is more important than ever to develop techniques for monitoring marine mammals that are cheaper, more reliable and more practical than traditional methods, to enable continuous monitoring to be possible. In Section 2.4.1, traditional methods of detection are reviewed with a proposed detection platform detailed in Section 2.5.

2.4 Detection platforms

Despite the highly threatened status of NARWs, the platforms used to detect and monitor populations have changed relatively minimally prior to 2010. Monitoring cetaceans is a challenging task due to an exceptionally inaccessible habitat and largely unknown behavioural patterns [18]. A combination of both robust detection methods and a reliable detection platform, aim to enable the best coverage of cetacean detection possible.

2.4.1 Traditional platforms

Traditionally, ship surveys have been the main source of information retrieval and often focused around sighting cetaceans [11, 23, 85] or recording their vocalisations via a ship-towed hydrophone array [236, 114].

For larger monitoring efforts, visual surveys involving aircraft flying over regions of ocean are common [23]. Although manned visual surveys have achieved some success for monitoring large regions of ocean [60], they, similarly to other techniques, have drawbacks limiting their detection and classification accuracy. From an aerial view, cetaceans are most visible at or near the surface, therefore visual surveys can be unreliable when cetaceans are deep-diving [137]. Cloud coverage, wave height and sun glare can all also dramatically affect initial detection results or classification validity [219, 137]. Furthermore surveying is restricted to daylight hours [137] which can significantly shorten the opportunity to report detections. Due to these limitations it is thought that some cetaceans in some circumstances, e.g. at night and during inclement weather are easier to hear than see [230] and consequently using acoustic detection would produce a higher number of detection events. In order to fully profile and monitor all aspects of a cetaceans behaviour, a combination of both visual and acoustic monitoring would provide the most insightful findings. However, for meeting the criteria of the NOAA recovery plan [62], acoustic monitoring is more suitable.

As previously mentioned, ships are a current and popular detection platform for monitoring NARWs. Surveyors use hydrophones in an array formation to record multiple streams of acoustic data [236]. A depiction of this process can be seen in Figure 2.3, showing a survey vessel towing a hydrophone array. This process has been named passive acoustic monitoring (PAM), where *"passive"* refers to listening for sounds in a non invasive manner [137]. Active monitoring is an alternative method of monitoring where surveyors use active sonar to search for their target [137], but this does not provide the insightful vocalisation information that PAM does.

PAM has gained popularity in recent years as recording equipment has become cheaper and more accessible. Furthermore, researchers now fully understand the benefits of acoustic recordings over visual only surveys. Sound transmission varies from water to air. However, water has the distinct benefit of being denser than air, which enables sound to travel further [163], providing acoustic detection from larger distances. Since many cetaceans are acoustically active [63], detection via PAM is the best way to gather information on the NARW and passive acoustic monitoring can be carried out continuously throughout the day and night with the correct equipment. Acoustic monitoring techniques do however have limitations, most notably differences in vocalisation behaviour and background noise variability [137]. Recorded data must also be stored or disregarded and since it is difficult, time consuming and expensive to obtain, it is likely to be kept and requires sufficient storage. Without an automated process for analysing the recordings, the data is often kept but not processed and further findings are more restrictive because of this. Detections are still noted, but further findings such as frequency, duration and amplitude of calls, can be missed unless time is spent listening to each recording.

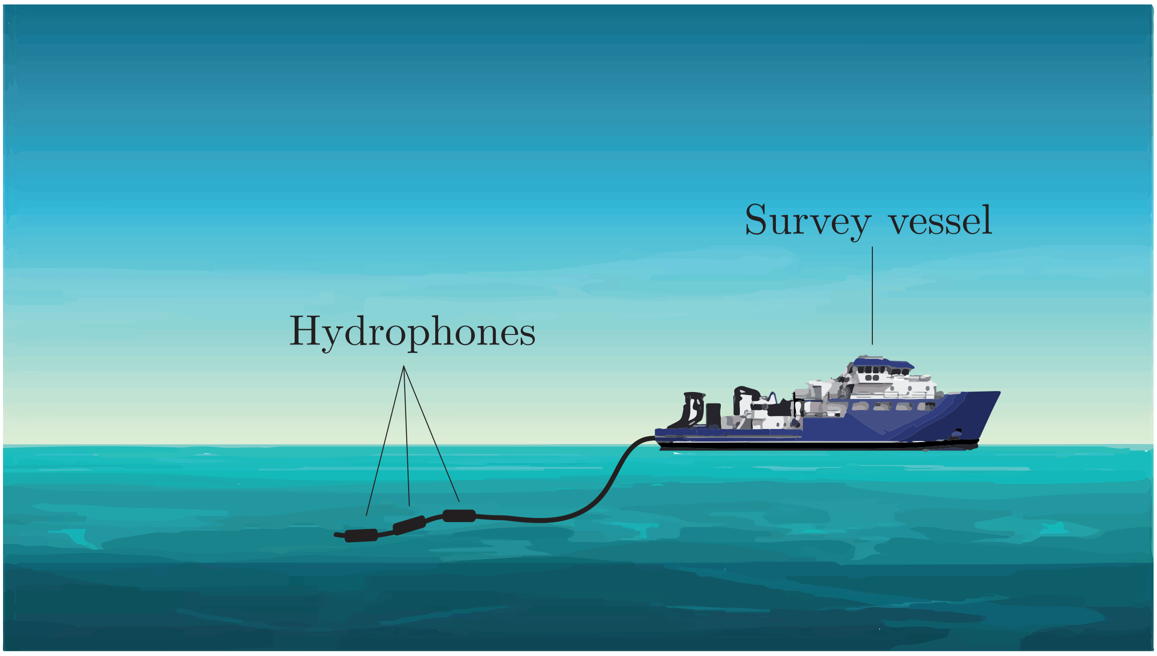


Fig. 2.3 A ocean landscape scene depicting a survey vessel towing a hydrophone array as would be found during acoustic surveys using ships.

Ships provide a platform for covering regions of ocean and recording data for observers to listen to. Since the visual and acoustic surveying drawbacks are mutually exclusive, observers often combine acoustic listening whilst visually scanning the ocean surface [236, 137] to provide a higher level of accuracy when classifying each detection event. Whilst ships provide advantages over aircraft-led visual surveys, such as being able to survey for longer periods, they also have constraints that restrict their output and make continuous monitoring an infeasible task. Ships are restricted by a number of factors with the main constraints being; weather that negatively impacts prevailing ocean conditions and probability of sightings, ability for the ship to travel to the given location (for example prohibitive environments might include ice), higher running costs, and inadvertent inclusion of ship related noises corrupting hydrophone recordings making detection more difficult [114].

A limitation of all monitoring techniques is the inability to be 100% accurate when classifying a species. For example, visual surveys can provide visual confirmation that a species is present, but require an acoustic detection to provide a higher degree of certainty when classifying vocalisations. Conversely, acoustic surveys can indicate the presence of a marine mammal but without a matched visual sighting the vocalisation label is an estimation of the vocalisation source. Therefore it should be noted that all ground truth can contain a degree of error.

2.4.2 Autonomous platforms

In more recent years there has been a dramatic increase in the number autonomous ocean vehicles [225, 211]. The invention of static buoys, ocean gliders, and surface vehicles has contributed to the rise of new platforms for monitoring and researching the world's oceans. Although still in their infancy, small ocean vehicles have gained significant popularity [18, 49, 136, 138, 29, 30, 31, 24] in the marine environment for their range of improvements over ship and aircraft monitoring. Ocean vehicles such as gliders and surface vehicles are relatively low cost in comparison to using ships or aircraft for monitoring. They can also operate in nearly all conditions, and consequently are able monitor continuously, even when prevailing conditions are too hazardous for human-based activity. They are mainly powered by long lasting batteries and surface vehicles can gain power from the sun using on-board solar panels [24]. Battery endurance can range but in ideal conditions can last for 3 months of continuous surveying before needing replacement [31]. The range of small unmanned vessels capable of PAM, are broadly categorised into three groups, although many variations of models and designs exists.

1. **Static buoys**

Static buoys, also known as moored buoys, are aimed at providing a clean recording of ocean sounds, but only in a few cases provide real-time feedback from an on-board PAM system [17]. This platform for detection uses a sea-floor mounted base containing hardware for recording and processing audio. The base is tethered to a buoy where communication hardware is positioned to send data to a shore-side computer via a satellite [17]. Figure 2.4 shows a depiction of a static buoy setup. Static buoys offer little operational noise in recordings and provide a stable location for continuous monitoring. As they are stationary they lack the ability to survey areas of interest, unless moved, and instead are better for a fixed monitoring environment.

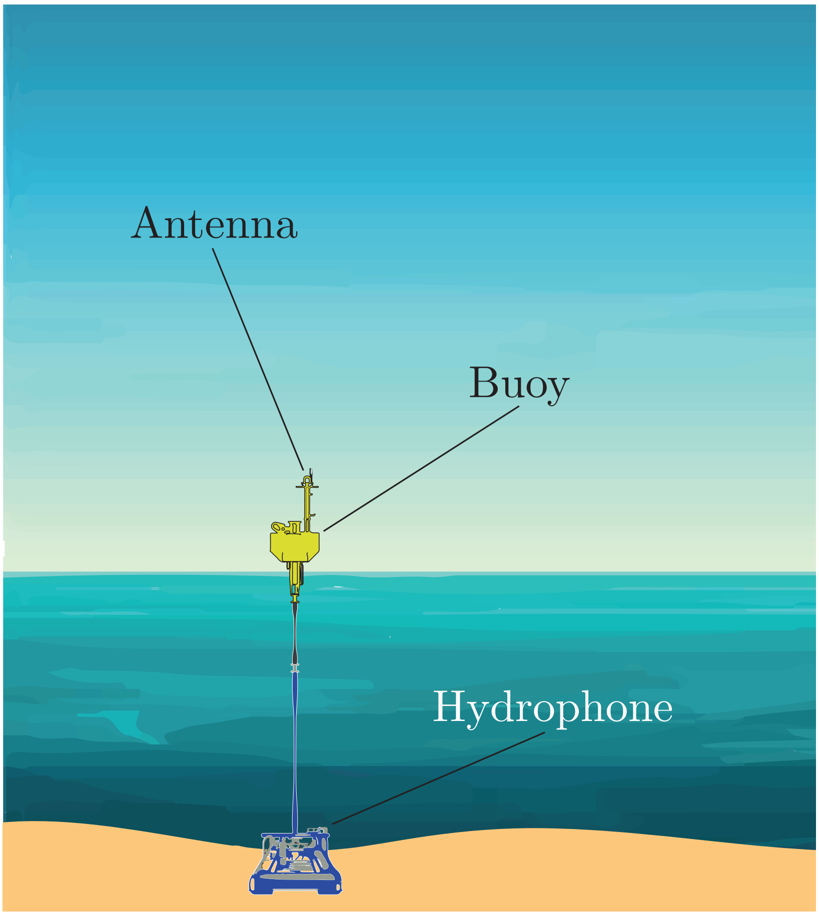


Fig. 2.4 An ocean landscape scene depicting a static buoy with a hydrophone and communication system. The buoy system is a reconstruction of Figure 1 in [17].

2. **Sea gliders**

In the context of this thesis, a glider refers to an ocean vehicle which primarily spends time below the surface of the water. Many variations of glider exists but one of the most common is a Seaglider [56], which was developed at the University of Washington. An ocean landscape containing a glider can be seen in Figure 2.5. The Seaglider uses a combination of battery movement and oil pumping to generate motion. Buoyancy of the glider is controlled by pumping oil into and out of a internal bladder, which induces vertical motion [31]. Guide rails shift the battery mass forwards and backwards within the unit to enable direction change in tandem with the buoyancy system. Seagliders are used for a range of scientific research and can be fitted with a fleet of sensors for varying applications [110, 171, 32, 15]. A PAM system can be attached to a Seaglider however noise introduced from movement of the battery and operation of the oil pump can increase noise in recordings. When the glider is either descending or ascending the water column, noise-free recordings can be attained as the internal components are stationary.

3. **Autonomous surface vehicles**

Autonomous surface vehicles (ASVs) are designed to navigate the surface of the ocean following mapped routes. They provide the benefit of continuous satellite communications link, which is also present on static buoys. Much like gliders they also have the ability to survey areas of ocean instead of remaining stationary. ASVs often carry out PAM similarly to ship surveying by towing a hydrophone array. Baumgartner et al. found that water flow and vehicle operation noise can appear in recordings, however with further investigation found that these can be mitigated with hardware alterations [16]. ASVs use batteries to power on-board equipment with certain designs incorporating small propellers, whilst others aim to glide on waves and use a rudder

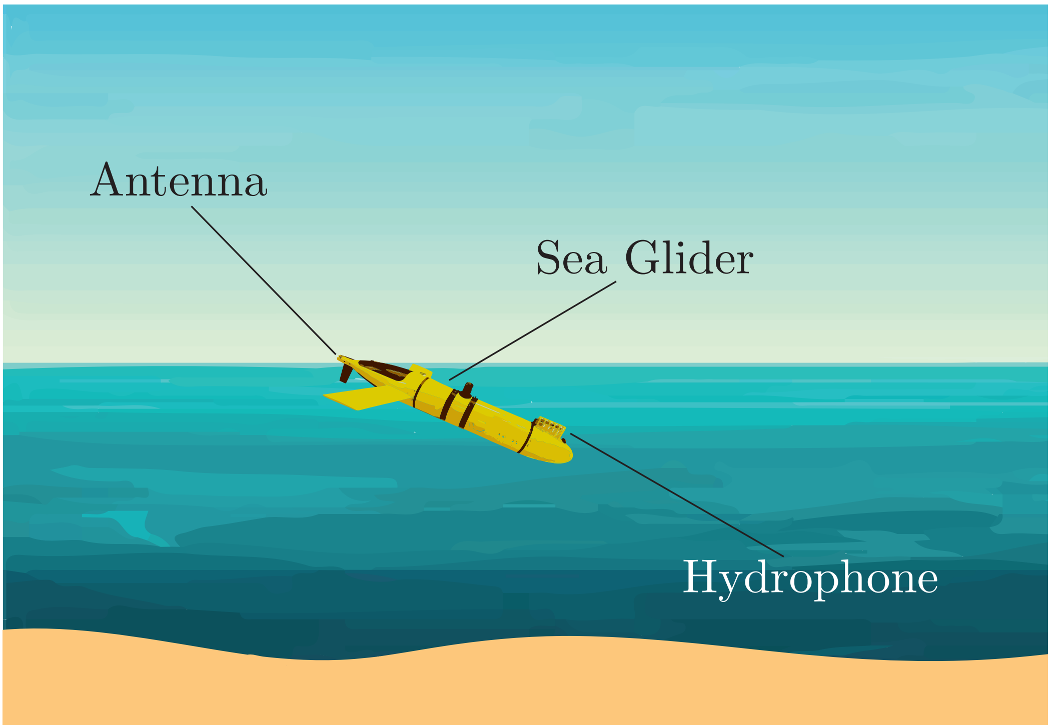


Fig. 2.5 An ocean landscape scene depicting a glider with a hydrophone and communication system. The glider is a reconstruction of an image taken from [180].

for direction. Another benefit of being surface level is the ability to use solar power to recoup energy lost. An ocean landscape containing an ASV can be seen in Figure 2.6.

2.5 Investigating autonomous platforms for North Atlantic right whale detection

Since autonomous platforms (APs) are able to combat many of the shortcomings present in traditional ship surveys, they provide a potentially more useful platform for NARW monitoring. APs however do have limitations, mainly because of their size. When autonomously navigating they can be a victim of oceanic drift, moving off path due to the ocean current or weather conditions. Whilst out for prolonged periods they can be a subject of biofouling, leading to the blockage of sensors and reduction in

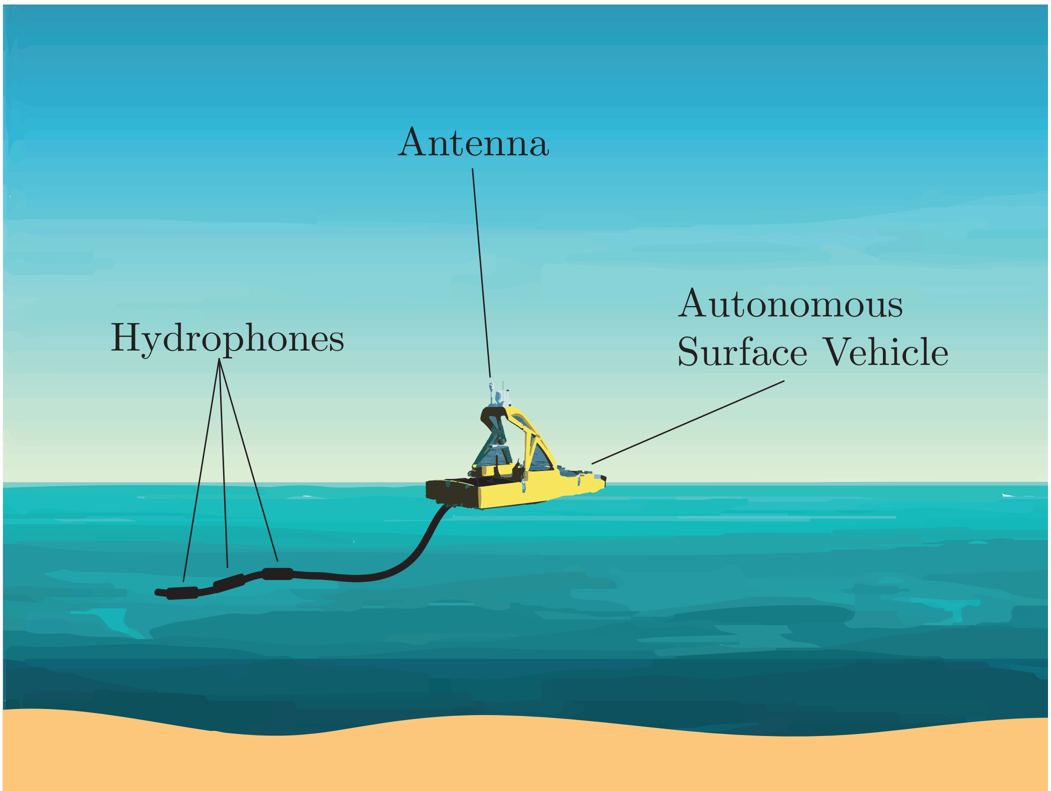


Fig. 2.6 An ocean landscape scene depicting an autonomous surface vehicle with a hydrophone and communication system. The ASV system is a reconstruction of an image taken from [200].

equipment effectiveness.

A common difficulty when dealing with APs is how to retrieve data from the AP. In ideal situations the platform would be retrieved after surveying and data collected. APs are fitted with communication devices to report back to the operator, but if this system fails, the AP can be hard to locate, leading to a loss of the platform, time and potentially addition of hazardous material to the ocean. Manually retrieving data also hinders applications such as NARW real-time mitigation alerts and therefore surface level APs with continuous connection might be more suitable. Due to the requirement of real-time monitoring, further work will only consider the use of autonomous surface vehicles (ASVs) and static buoys. Due to the nature of ASVs and their surface presence they can house communication equipment that can maintain a continuous link with land-based systems, unlike gliders which can only connect when resurfacing.

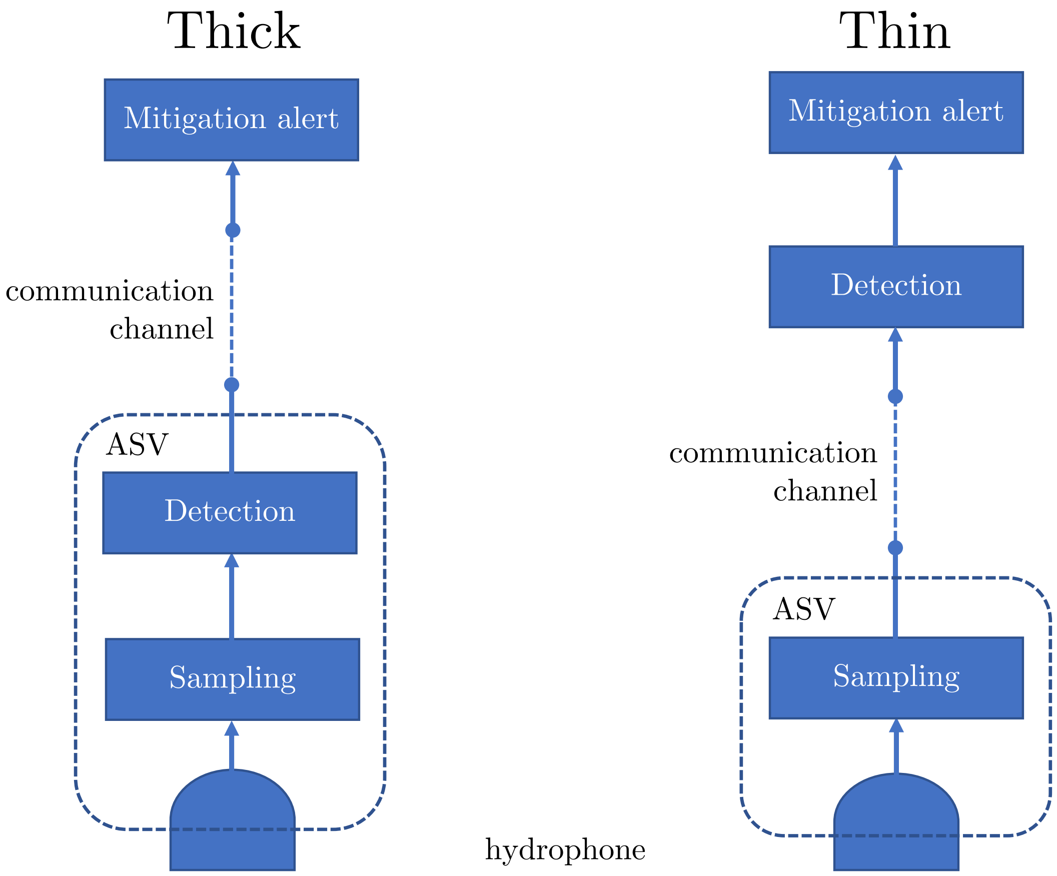


Fig. 2.7 Two proposed approaches for NARW detection systems using PAM.

Two methods for using ASVs as a detection platform for NARW detection are now proposed. Both methods can be applied to static buoys, however for simplicity only ASVs will be discussed. Figure 2.7 shows both the “thick” approach on the left and the “thin” approach on the right. The “thick” system is a self contained detection platform providing real-time detection on-board the ASV. Using the on-board computer the “thick” system is designed to read in audio, processing a continuous stream. Detection occurs during this process and results are send via satellite to the shore-based receiver. The “thin” system operates a slimmed down pipeline with no detection occurring on-board, instead audio is transmitted to land via satellite for detection. Both systems provide various advantages, for example the “thin” approach is more lightweight without onboard detection, requiring less power to operate and can therefore survey for longer periods with the same battery capacity as the “thick” system. The detection process for “thin” approach can also be refined and updated without disruption of the survey. However, the “thick” system is the only reliable method of detecting NARWs and receiving notification in near real-time as satellite links are expensive, potentially unreliable and have slower transfer speeds, sending minimal data is preferred. The “thick” system would only need to send notification of a possible detection for land-based mitigation alerts to be triggered, whereas the “thin” system would require a stable connection to transmit continuous audio, costing more in transmission fees and potentially being unreliable in unstable network conditions. A concern of the “thick” system is the necessary computing power needed to run a robust NARW detection algorithm in real-time on a system which draws minimal power. This work will aim to investigate the use of low-powered computing for running a complete detection system in real-time and will consider not only detection accuracy but also processing constraints.

2.6 North Atlantic right whale vocalisation data

As previously discussed, labelled recordings of marine mammals are hard to attain without expensive equipment and expert domain knowledge of each specific marine mammal. For this reason, data collection does not form part of this work and instead the data used within this thesis has been collected from existing sources. A breakdown of all datasets used is detailed within this section.

2.6.1 Vocalisation behaviour

The vocalisation behaviour of NARWs is an important factor to be considered when attempting to understand and detect their vocalisations using PAM. The type of vocalisations an animal produces, the acoustic and time frequency at which they are produced and their call characteristics are all factors which have the ability to influence how detection is carried out. It's currently thought that NARWs do not sing and sequences of calls can therefore be non repetitive and irregular [39]. Although knowledge of NARW call occurrence is sparse it is suspected that call density is low, with only 690 calls recognised in 300 hours of recordings [133]. From this it is not possible to reliably establish call rates, however a frequent repetitive calling pattern is highly unlikely.

North Atlantic right whales are known to produce a number of vocalisation with the upcall (Figure 7.1 left) and gunshot (Figure 7.1 right) sounds the most common. Upcall vocalisations are produced in the frequency range of 50-400Hz [158] and are typically seen as a sweep up in frequency over time. The upcall typically has a duration of approximately 1 second. The second most common NARW vocalisation is the gunshot. The gunshot is a high amplitude broadband signal which has a duration of approximately 0.5-1 second [157].

2.6.2 Assessing performance

When aiming to classify vocalisations of NARWs the metrics used to assess performance of the underlying classification system must be suitable in order to present meaningful results for evaluation. A number of techniques are used to report the success of a classification system with accuracy being the most common. In the field of marine biology, classification metrics such precision-recall curves, receiver operating charac-

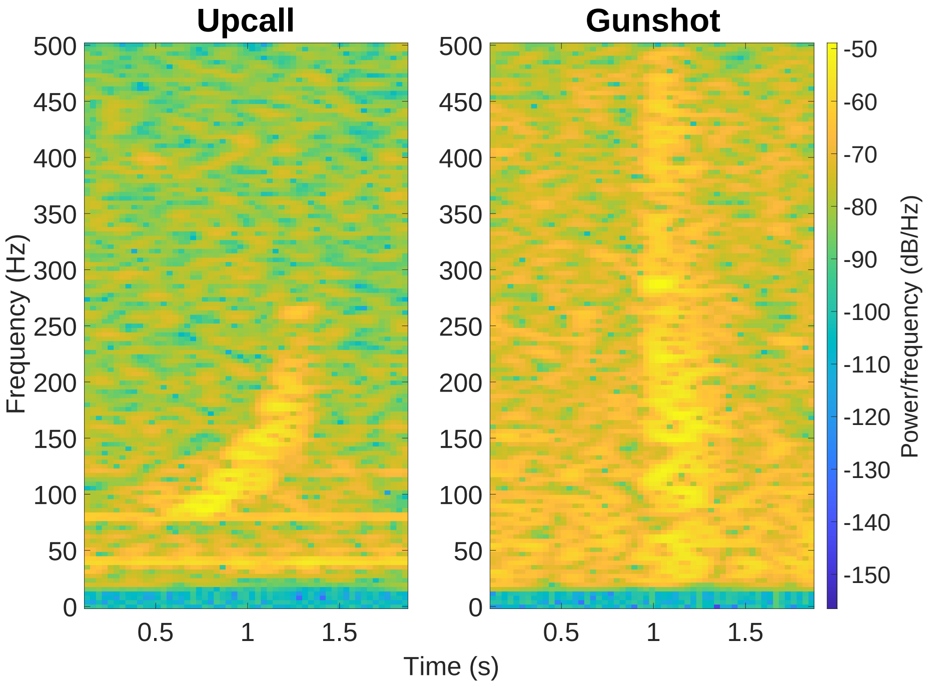


Fig. 2.8 Two example spectrograms showing a right whale upcall (left) and a gunshot (right). Upcalls are characterised as a tone starting at around 50Hz and ending around 400Hz, with a duration of one second. Gunshots have less structure and are characterised as bursts of broadband noise.

teristic curve (ROC curve) and Area under the ROC curve (AUC curve) are all also frequently used as they give further insight into the confusion matrix of a classifier instead of evaluating overall performance as is shown with accuracy. Throughout this work accuracy will mainly be used evaluate performance as this enables test to be more directly comparable to each other, however for further analysis of experiments in later sections both precision-recall and ROC curves will be used.

Although the accuracy metric is mainly used throughout this work, the importance of other metrics is greatly understood. As previous work has shown the NARW to infrequently vocalise [133], it becomes important to evaluate potential classifiers with methods that accurately replicate real-world scenarios. Awareness of where the classifier may be embedded is important factor when designing a full classification system. An example of a more relevant metric might be the number of false alarms per hour as shown in [188], where incorrectly identifying NARWs may lead to serious time and monetary consequences, such as the closure of shipping lanes or shut down of off-shore construction sites. When considering the false alarms per hour, context of the current marine mammal may also be crucial when setting an acceptable alarm threshold when developing the system. For example as the its not always critical to classify every upcall correctly, the classification threshold could potentially be lowered to meet the acceptable false alarm. Although integration of wider context appropriate metrics is important, for the fundamental investigation and design of the classifier, the work within this thesis focuses solely on using the accuracy metric for comparable experiments.

2.6.3 Available datasets

Throughout this thesis three datasets have been used for experimentation work. Below is a dissection of the available datasets to show where they came from, their size, and how they have been manipulated before use.

1. **Cornell**

The *Cornell* dataset refers to The Marinexplore and Cornell University Whale Detection Challenge [206] posted on Kaggle, a machine learning competition website. This dataset was provided by Cornell University for a competition to detect NARWs within audio segments. The dataset is freely available to download however the competition closed in 2012. Provided alongside the competition dataset were training and testing segments of NARW detected events containing the most common NARW vocalise - an upcall. All training segments had designated label files to indicate whether a NARW upcall was present or not. The aim of the competition was to generate subsequent label files for the remaining test segments and to submit predictions to be judged. As the competition has closed, the test segments were redundant because related label files were not accessible and therefore only the training segments were utilised within the Cornell dataset. Within the training folder, 30,000 segments were present. 22,973 were labelled as *"not-NARW"* with the remaining 7,027 upcalls labelled as *"NARW"*. It should be noted that the *"not-NARW"* class has the potential to contain anything that is not a NARW upcall and therefore may contain other marine mammal sounds, shipping noise or other ocean sounds. Initially the dataset was manually restricted to a maximum size of 14,054 segments as this maximised the *"NARW"* segments and gave an equal class split. Analysis of the segments' energy content found some to have vastly different energies, so these were removed for consistency. After extraneous segments were removed, 14,016 segments remained, these were distributed equally across *"NARW"* and *"not-NARW"* and were split 70:15:15 for training, validation and test, this breakdown can be seen in Table 2.1. All files were shuffled prior to splitting to ensure all timesteps were mixed in case the original set were given in chronological order. All segments are presented as 2-second duration blocks of audio, sampled at 2kHz.

Although this dataset was collected from Kaggle where a number of competition entries have been made, a direct comparison to results in this thesis cannot be established as the labelled test data was unavailable to post-competition users. Entrants of the Kaggle competition had access to 30,000 labelled training segments with a further 54,000 used to test their classifier. This work only considered a subset of the original dataset and instead only used 12,008 labelled training segments with 2,008 reserved for testing. As the training sizes are significantly different, comparison between competition entries and this work have not been made. Results between competition entrants and the experiments presented here are also provided using different metrics and therefore are not directly comparable.

Dataset Name	Train	Validation	Test	No. Classes
Cornell	10,000	2,008	2,008	2
Cape Cod	10,000	2,142	2,142	2
Stellwagen Two	10,000	1,690	2,142	2
Stellwagen Three	2,784	600	600	3

Table 2.1 A table showing the number training, validation and testing segments available to each dataset used. Classes were evenly distributed for each dataset.

2. **Cape Cod**

The *Cape Cod* dataset refers to a set of ∼160,000 audio segments procured from a NARW monitoring website [205]. Although authorship is lacking, is it presumed that this service is owned and run by Cornell University bioacoustic department. The website uses a series of 10 statically moored buoys to report NARW upcalls. This is a live service and reports NARW upcalls in near real-time with each recorded segment, corresponding spectrogram and buoy information available to view shortly after detection. As the audio files are freely available to download, a script was written to continuously pull files from the server. The files were downloaded with an equal split between confirmed *"NARW"* upcall segments and rejected *"not-NARW"* segments. This service provides rejected files that mostly contain acoustic events, but have been rejected as being NARW upcalls for which are then used as *"not-NARW"* within this dataset. The labels are originally produced after detection by a frequency contour algorithm [69] and manually authenticated by a human operator. Cape Cod may therefore encompass a wider range of bioacoustic events. Table 2.1 gives a detailed breakdown of the 70:15:15 split of training, validation and testing segments. Originally a larger corpus of 100,000 training samples was used as it offered a substantial amount of additional data when compared to other datasets. However, initial tests exploring training set volumes, discovered that using more than 10,000 training samples provided a minimal gain in accuracy and therefore a smaller subset of 10,000 training samples was taken and used for further tests. Cape Cod uses a portion of the full download of data and provides a more reasonable amount of data to process and test. Similarly to Cornell, all files were shuffled prior to splitting to ensure all timesteps were mixed in case the original set were given in chronological order. All segments are presented as 2-second duration blocks of audio, sampled at 2kHz. Both Cornell and Cape Cod datasets have originally been detected using a low threshold amplitude detector [69]. The classification system designed in this work uses the original detector labels with the aim of producing a superior classification system.

3. **Stellwagen**

The *Stellwagen* dataset is a subset of the Detection, Localisation, Classification and Density Estimation (DCLDE) 2013 conference [193] competition data. DCLDE 2013 hosted a workshop competition and made a dataset of NARW calls available for participation. Although the conference was in 2013, the dataset and website have remained active since. The data was collected using marine autonomous recording units (MARUs) deployed in arrays of between 6 and 10 devices off the North Atlantic coast at Massachusetts, US. For this dataset, the output of just one channel is taken, converted to 16 bits per sample and sampled at 2kHz [193]. The audio recordings have been annotated by human experts using data from all channels to maximise accuracy [193].

Similarly to the Cornell dataset, training and testing files were available, however associated log files for the test data were not available so only the training files make up the Stellwagen dataset presented here. Unlike the previous datasets the competition made available NARW upcalls and NARW gunshot sounds. Data with no NARWs was also provided. All recordings were given at a sampling frequency of 2kHz, with files presented as 15-minute recordings, spanning multiple days worth of continuous monitoring. Log files were made available and contained a detailed breakdown of detection events, giving the start and end time, and lower and upper frequency of each detection. In order to match this dataset to those previously used, acoustic detection events were removed from the larger files, centered and padded to give a 2-second block of audio for each event. It should be noted that padding uses audio from directly before and directly after the designated event window. Initial analysis of the event files found both upcalls and gunshot files to often contain noise corruption from a low frequency stationary source, that is likely caused by the sound of a mechanical hard drive spinning up. This corruption caused visible horizontal banding in the spectral-domain. Although a set of dedicated set *"not-NARW"* files were available, these did not contain the same corruption as seen in the upcall and gunshot files. In order to combat a classifier simply learning the difference between the noise corrupted and non-corrupted recordings, the decision was taken to extract *"not-NARW"* segments from both the upcall and gunshot recordings. Instead segments labelled as *"not-NARW"* were taken 5 seconds after a real detection event to ensure similarity in background noise. If there was less than 5 seconds between events then the next available gap was used as a *"not-NARW"* segment.

As gunshots and upcalls were labelled in the dataset, two Stellwagen combinations were created - *Stellwagen Two* and *Stellwagen Three*. Stellwagen Two matches that seen previously and contains two classes, one for *"not-NARW"* and another for *"NARW"*. Stellwagen Three utilises all available classes, and is made up from {upcall, gunshot, not-NARW}. A breakdown of the training, validation and testing splits can be seen for both datasets in Table 2.1. For consistency, the number of Stellwagen training segments was matched to that of both the Cornell and Cape Cod datasets. As less detections were available for Stellwagen, the number of test segments were kept consistent with Cape Cod and roughly Cornell, however a reduction of validation segments was necessary. All splits contain an equal proportion of each class and were shuffled when finalised.

2.6.4 Marine and environmental noise

Given the nature of cetacean monitoring, often recordings can be masked by a range of anthropogenic sounds. Noises from ships, fishing trawlers, or military testing can all contribute to noisy recordings making detection of NARWs significantly more difficult. Many datasets provide clean recordings with a high signal-to-noise ratio (SNR) to make training machine learning models straightforward. These datasets are also the most accurate as the bioacoustic signal of interest can be clearly heard and often matched with visual confirmation in the spectral domain. When noise is introduced, detection and classification becomes more challenging and class labels are less reliable. PAM recordings are seldom clean and therefore datasets containing majority clean recordings are not a true-to-life representation of the natural world.

To combat dataset bias, four marine-based noises have been collected and added to the Cape Cod and Stellwagen corpus, presented in Section 2.6.3, to simulate real-world noisy conditions. The four noise types considered for the evaluation are; i) tanker noise, ii) trawler noise, iii) shot noise and iv) white noise. Spectrogram examples of each of these noise types are shown in Figure 2.9. Tanker and trawler noises were chosen, as shipping is a common source of marine noise that introduces horizontal bands in the spectrograms arising from harmonics of rotating machinery within the ship and low-frequency noise. Additionally, fishing trawler nets and boat strikes are the leading cause of death and injury to cetaceans; thus, the noises considered are likely found in mitigation zones, such as shipping lanes, construction sites and designated fishing areas. These noises were obtained from data that had been collected by the NOAA Northeast Fisheries Science Center from a passive acoustic monitoring project in the Stellwagen Bank National Marine Sanctuary. Shot noise is representative of sounds produced by activities such as piling and seismic exploration and is characterised by a vertical structure in the spectrogram. This noise provides likely environments for mitigating NARW injury in areas surrounding construction or deep-sea exploration. The shot noise examples were taken from the 'gun' samples in the NOISEX-92 database [210]. This noise is impulsive but was arranged so that each two-second recording contained at least one example of the shot noise. Finally, white noise is included as a more general noise type that affects all time and frequency regions within the spectrogram, and this was generated artificially. White noise provides an example of a generally noisy ocean environment, possibly where multiple noise sources are corrupting PAM recordings simultaneously. As white noise causes corruptions across the entire spectrogram it also provides a challenging condition in which noise reduction methods can be effectively tested. For consistency, all tests used the same noise segments for corruption. One set of white noise segments were created and the same segments were applied to any tests using white noise. The same segments were used for training, validation and testing for each test.

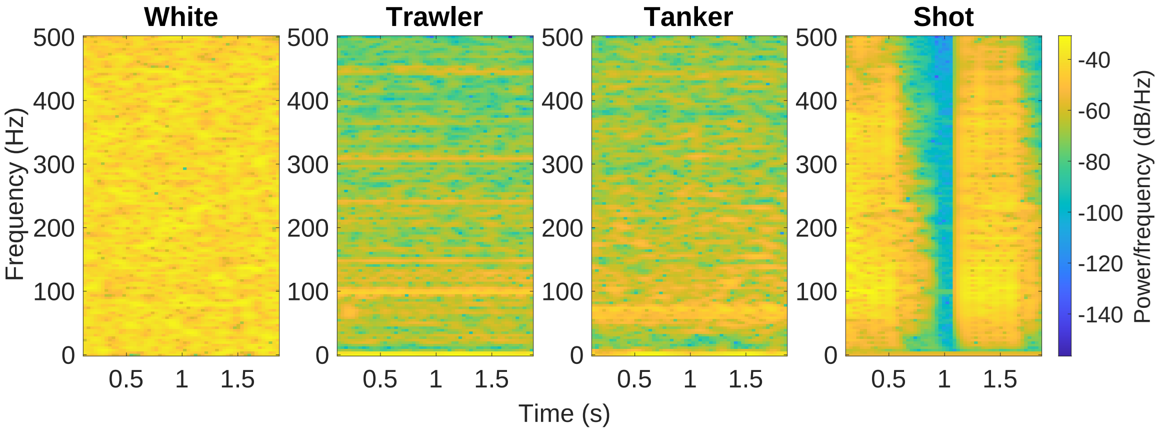


Fig. 2.9 Spectrograms showing two second examples of white noise, trawler noise, tanker noise and shot noise that are used in the evaluations in Chapter 7.

To create the noisy audio segments, noise is added to every two-second recording in the time-domain at SNRs of 5dB, 0dB, -5dB and -10dB. This set of SNRs is chosen to cover a range of reception conditions that represent signals received from NARWs at both close and long range distances. For recordings that contain a NARW vocalisation, the noise samples are scaled such that when added to the NARW recording, their subsequent power achieves the target SNR.

To add noise to the *"not-NARW"* recordings, 2-second segments with no NARW vocalisation present are extracted from the original recordings at a time 5 seconds after an upcall or gunshot has occurred. To these *"not-NARW"* segments, noise samples are added and scaled so that they have the same noise power as that in the preceding segment which contained a NARW vocalisation. This ensures that the actual power of the noise remains consistent across each pair of *"NARW"* and *"not-NARW"* examples. The procedure is illustrated in Figure 2.10. For the context of this work, $\mathbf{x}$ is assumed to be a clean, non-noisy signal with, $\mathbf{d}$ representing noise. As shown in Equation 2.1, $\mathbf{x}$ and $\mathbf{d}$ are added to create a noisy signal, $\mathbf{y}$. Both the signal power and noise power are calculated using Equations 2.2 & 2.3 respectively and shown in Figure 2.10. The noise scalar shown in Figure 2.10 is represented by α in Equation 2.4 & 2.5. The noise scalar controls the level at which noise is added to the signal, and enables the noise to be added at the desired SNRs. It should be noted that within a specific SNR and noise type, the noise examples that are added to the *"NARW"* and *"not-NARW"* examples are not duplicated, so each 2-second segment is contaminated with unique noise examples. Further, there is no sharing of noise examples used across the training, validation and testing sets. All noisy dataset variations are clearly defined in text. If there is no noise mentioned then the original non-noisy dataset is used.

It should be noted that adding noise to a dataset in the above manner is not without fault. Automatically selecting segments of audio targeted at being *"not-NARW"* 5 seconds after a detection event does not ensure that the segment is noise free but instead each segment has the potential to be corrupted with anthropogenic sound; or cause further confusion and contain an alternative mammal vocalisation. To combat this, the *“not-NARW”* is specifically labelled as such to present the understanding that samples may contain anything that is not produced by a NARW.

Two problems are understood to be present when detecting NARWs; firstly, a continuous stream of ocean recording is available which requires events to be detected prior to classification; and second, classification of detected events. This work and the results presented, explores the problem of NARW classification as a post-detection process, at which point each detection event has been determined for further processing to establish the class of the event.

$$\mathbf{y} = \mathbf{x} + \mathbf{d} \tag{2.1}$$

$$P_{signal} = \frac{1}{N} \sum_{n=0}^{N-1} \mathbf{x}(n)^2 \tag{2.2}$$

$$P_{noise} = \frac{1}{N} \sum_{n=0}^{N-1} \mathbf{d}(n)^2 \tag{2.3}$$

$$\alpha = \sqrt{\frac{P_{signal}}{P_{noise}} 10^{-\frac{SNR}{10}}} \tag{2.4}$$

$$\mathbf{y} = \mathbf{x} + \alpha\mathbf{d} \tag{2.5}$$

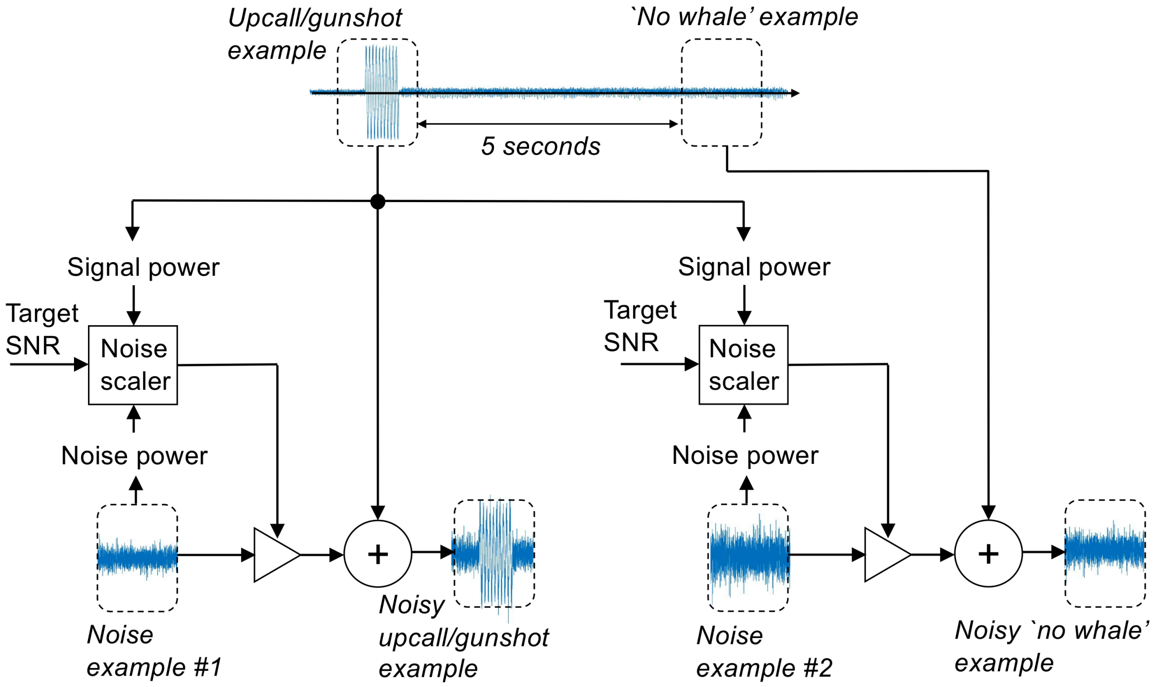


Fig. 2.10 Method of adding noise to the two-second “NARW” and “not-NARW” segments to create noisy examples at the target SNR. “Not-NARW” examples are extracted 5 seconds after a NARW vocalisation to give consistency in terms of the power of the noise examples that are added.

2.7 Traditional detection techniques

This section covers early work implementing methods of detection that were more traditionally used for cetacean detection, prior to the rise in popularity of time series classification and deep learning algorithms [109, 199, 226, 4]. Machine learning has become popular for detection and classification in part due to the growth of data collection and also the availability of high performance computers. Previous to the use of machine learning, methods to detect cetaceans were largely based on signal processing methods. Without excessive amounts of data to build complex models, techniques relied on traditional signal processing operations such as filtering and masking to make detection easier. This section explores the use of a traditional detection approaches and explains methods to analyse ocean recordings when ground truth labels are not available.

2.7.1 Amplitude thresholds

One of the most fundamental functions to detect the presence of a signal within a block of audio is to use an energy threshold detector (TD). A TD is a method of reporting energy values within a signal that cross a given detection threshold, *TH.* Reported energy values provide samples for when the signal exceeds a threshold, which indicates that an acoustic event may have occurred. If a segment of audio has an energy which significantly exceeds the background noise energy, then the segment is considered a detection. Methods that threshold an acoustic signal do not necessarily need to operate in the time domain and instead can use features such as spectrograms [69].

Specifically, a TD works well to detect transient calls often seen when monitoring cetaceans that produce echolocation clicks [229]. However this method can be applicable to all cetaceans when the SNR of a vocalisation is such that the threshold is past.

Signal processing filters deliver a reliable method to attenuate frequencies within a given frequency range and can be used in combination with a TD. Prior to using a TD, the acoustic recording can be filtered to produce a signal with a limited frequency range, reducing noise and concentrating on frequencies known to contain the cetacean sound. Filtering in conjunction with using a threshold can yield suitable results for detecting specific cetacean vocalisations. NARWs however produce upcall vocalisations in the 50Hz-400Hz band and therefore are difficult to detect, using this method, unless the SNR is high. Although, filtering provides the ability to mask higher frequencies which NARWs are known to not vocalise at, which can therefore reduce confusion within the detector. [69] found some success for NARW detection using a threshold on a smoothed spectrogram, however in more noisy condition such a technique may be inadequate to effectively threshold the NARW vocalisation for detection.

2.7.2 Clustering

Clustering is a unsupervised method of classification that relies solely on the input data. Attributes of the input data can be extracted and deemed, ‘features’, which enable the clustering algorithm to operate. Initially two false means can be created and then centered based on which detections features are closest. This can be repeated until a desired number of clusters have been created. Calculating standard deviation and variance of clusters can help to analyse the spread of each cluster. Attaining a small variance can indicate that clustering might be complete as data points are closely matched. Data in a more compact region is likely to be similar in nature and therefore created from the same source. As an example, NARW vocalisations could be clustered by event duration, event fundamental frequency, and event energy. Accuracy of clustering does however rely heavily on the reliability of the extracted features. In noisy conditions the reliability of features may become uncertain as noise may artificially increase a feature such as event energy or change the fundamental frequency. Once clusters are formed, each event is labelled with the cluster that it sits closest to. Self generated labels from clustering can later inform further supervised algorithms. This pipeline would be entirely unsupervised and may produce initial classifications without the need for labelled data.

2.7.3 Alternative methods of detection

A range of methods have been explored in the past as techniques for detecting marine mammals. Mellinger and Clark [140] developed an automatic recognition method for transient whale sounds. They use spectrogram correlation with the aim of mapping distinct spectrogram contours using a recognition score function. Higher peaks within the recognition score function indicate a correlated spectrogram and thus a detection event. Results found the correlation function to perform better than more simple matched filter function, however they acknowledge that neural network methods can achieve higher accuracies. The combined error rate of the spectrogram correlation method was 2.5% with a neural network achieving 1.6% although significantly more training samples were used. Performance in noisy conditions with a low SNR however, was not tested.

In 2006 and later in 2009, Urazghildiieva and Clark [207, 208] used a generalised likelihood ratio test (GLRT) to evaluate spectrograms of NARW calls within background noise. They found that using GLRT to detect NARW upcalls worked reasonably well with some high detection probabilities but this often came at the cost of an increased false alarm rate. In noisy conditions the results are seen to worsen and it is suggested that less due to a lack of *a priori* statistical information, detection becomes increasingly difficult.

Previous work was then superseded by Baumgartner [14, 18] in 2011 and 2013 with research aimed at addressing baleen whale detection by developing software called the low frequency detection and classification system (LFDCS) to detect low frequency whale vocalisations. The software first smoothed spectrograms of acoustic events using Gaussian smoothing kernels and attempted to remove tonal noises produced by ships by subtracting a low-duration mean from each spectrogram frequency band [14]. Potential calls are first detected using a threshold detector as described above, which are then mapped along their fundamental frequency to generate pitch tracks. Using each pitch track, a set of call attributes are established with these being, “start frequency, end frequency, frequency range, duration, and slope of frequency variation” [14] and this informs the final prediction. Call attributes can then also be used within a clustering algorithm and can help to form natural groupings within the data. Results for the LFDCS system were promising with performance similar to a human operator.

2.8 Discussion

This chapter has covered the phylogeny of cetaceans and explored why detection of NARWs is a vital task, and how this is currently carried out. A new detection system is proposed in Section 2.5, and aims to address many of the limitations present in current detection systems.

Available data and their sources have been outlined in Section 2.6.3, with analysis of common noise corruptions discussed in Section 2.6.4. In particular, this work considers sources of noise that are know to be produced by scenarios harmful to NARW. Once detection platform limitations are minimised, focus quickly shifts to the reliability and robustness of the detection method. As previously discussed, there is often large differences between dataset recordings, often with little background noise, and real-world recordings, with potentially a wide range of background noises. In order to combat a mismatch between the experimental and real-world scenarios, oceanic noise corruptions have been added to the original dataset segments to simulate conditions more similar to those of the real-world. Four specific noises were chosen which aim to represent a range of sounds that could be found in the real-world conditions, but also to cover a wide range of frequencies, causing varying amounts of disruption in the spectral domain.

One of the largest motivators behind building a robust noise tolerant detector, is to ensure monitoring efforts are equally as valuable when operating near shipping lanes or fishing trawlers. The noises chosen aim to cover a range of environments and represent the harsh conditions that NARW might be in when requiring the highest level of protection against harm.

Chapter 3

Development of machine learning methods for classification of North Atlantic right whale vocalisations

3.1 Introduction

This chapter is concerned with developing and comparing machine learning (ML) techniques for the classification of NARWs from an acoustic source. Machine learning is the process of learning and adapting an initial set of model parameters [146] from a set of training data. Continual learning, without following explicit instruction [177], consequently informs further adaptation of the machine learning algorithm with the aim of most accurately predicting which class future unseen data falls into. Classes in the context of machine learning refer to groupings within the data and are often manually defined prior to training. For example, for North Atlantic right whale classification two classes are used, "*not-NARW*" and "*NARW*". Prior to running a supervised ML algorithm, each block of audio will be labelled manually to define which class it belongs to. Supervised machine learning algorithms build a model based on a subset of the data, known as training data. The training data defines data samples for which the algorithm can see over and over again during a ‘training’ phase. Model training aims to improve the separation of classes over time and subsequently improve algorithm accuracy.

The aim of this chapter is to develop and compare a range of machine learning methods for detecting NARW vocalisations [214]. The machine learning methods investigated are broadly categorised into two groups; deep learning and time series. Multiple algorithms from both categories are explored and developed before finally comparing their performance on a dataset of NARW vocalisations. The deep learning algorithms all use variations of neural networks to build hierarchical architectures for processing the acoustic data, with each method fundamentally operating in a different way. The time series methods cover a broad range of methodologies in order to find the algorithm most suited to this application. Some of the methods use the raw time-domain signal without prior processing whilst others first extract features to use for classification. Investigating a wide range of machine learning methods with differentiating properties allows for an extensive survey of results and informs the classification method for future investigations and testing.

Throughout this chapter, both classification accuracy and processing requirements are considered in order to outline an optimum solution for classification via relatively low-cost and low-powered hardware such as an ASV. Given the correct hardware and classification setup, the aim is to process acoustic recordings and predict class labels in real-time, allowing for the system to run continuously without the need for manual interference.

The remainder of this chapter is organised as follows. Section 3.2 reviews previous methods of ML based classification, with the aim of exploring methods that have previously been applied to the problem of acoustic classification and also to more specialist methods used for NARW classification. Section 3.3 evaluates deep learning techniques and investigates feature extraction to provide the classifier with the most suitable input. Section 3.4 explores a range of time series based machine learning techniques and compares their success on NARW classification. A comparison between the best performing time series and deep learning methods is presented in Section 3.5, providing discussion and potential avenues for further exploration.

3.2 Background

This section aims to first provide a explanation of terminology used within machine learning for clarity in future sections. Next, deep learning methods are introduced with the first widely accepted deep learning architectures discussed. Time series classification methods are then discussed to explore their success in past classification problems. Finally, work specifically investigating machine learning for NARW classification is reviewed to understand what has previously been explored.

Machine learning is a term for incorporating a broad array of algorithms within the field of computing science that aims to understand the structure of data and build models that best represent that data for future classification of similarly presented data. Traditional algorithms operate under an explicit set of instructions to solve a specific task. For example a sorting algorithm will carry out the same operations repeatedly until finished. Instead, machine learning algorithms have been developed to use a learning period to update their internal model based on *seen* data. This learning period enables the algorithm to *train* a model to best represent the given data. Learning without following explicit instruction [177] defines these algorithms as enabling a machine to learn, hence the given name, machine learning. Recently the use of machine learning algorithms has grown exponentially with uses found in most modern computing applications [100], for example; facial recognition [156], speech recognition [162], and autonomous driving [65].

Two widely adopted training methodologies for machine learning algorithms are supervised learning and unsupervised learning [68]. Supervised learning entails learning a mapping between a set of input variables $\mathbf{x}$ and an output variable ψ and applying this mapping to predict the outputs of unseen data [28]. This mapping is learnt during a *training* period, when the algorithm calculates the difference between ψ and the ground truth label, g to find mismatches and update the model accordingly to reduce the error. Using a supervised model therefore requires, g to be present before the training period can begin. Often obtaining ground truths can be difficult and time consuming, making supervised learning potentially more costly [231]. However, the benefits of using supervised algorithms generally outweigh their cost, due to their reliability and structured learning approach, learning from real world truths [93]. Unsupervised learning equally tries to learn a mapping between $\mathbf{x}$ and ψ, however without knowledge of g [68]. Without prior knowledge of which class the data belongs to, the algorithm must instead approach the problem by learning relationships within the data. For example, a clustering algorithm would learn natural groupings within the data from data features [68]. Unsupervised learning approaches offer a chance to gather data insights without needing to collect ground truth information. For certain applications this approach is fundamental, for example when NARW ground truths are missing or inaccurate, an unsupervised approach might be the only possibility of building a classifier. Since the NARW dataset being used has ground truths available, further investigation will explore supervised learning methods.

Machine learning has become a popular solution for many applications in recent years [45, 104, 34] and gains popularity each year [100]. A reduction in computer component size [185] has lead to a surge of low-cost solutions for data collection [185] and thus methods to process the data and extract the most valuable findings has been an area of intense research and development. Progression frameworks such as the Internet of Things (IoT) [172] has also been responsible for making ML solutions more suited to larger audiences due to the always-connected nature of devices.

Deep learning (DL) as a subject area has advanced considerably over the last 15 years with popularity starting to grow in 2006 [88]. Although early neural networks date back to 1957 [173], the adoption of such techniques has only occurred more recently with the advent of deep learning, specifically focusing on creating deeper neural networks [111]. The deep learning family of algorithms has evolved from feed-forward networks, to recurrent networks, and convolutional networks. Research in this area is highly active, with new methods being introduced continuously. Each evolution has bought a different methodology whilst utilising the familiar underlying hierarchical architecture to solve a range problems.

The basic idea of a single perceptron (or node) was created in 1957 by Frank Rosenblatt [173]. Although useful at solving simple problems, they were restricted in design as they were incapable of learning the XOR function [145, 164]. The multilayer perceptron (MLP) solved this problem and provided the basis for fully connected neural networks that are used today [81]. Development from MLPs led to the creation of many offshoot networks. Recurrent neural networks (RNNs) were first detailed in 1986 [175].

RNNs were designed to help propagate information further down the network whilst back-propagating errors in order to learn complex representation of the data. Tasks such as speech recognition benefited hugely from this type of recurrent structure. Convolutional neural networks (CNNs), developed simultaneously, focused on image recognition with LeCun et al. in 1994 providing the first mainstream solution [209].

Despite early proofs [173, 84, 81], widespread deep learning adoption did not occur until 2010s when deep learning approaches started to outperform other methods. In 2013 the MNIST handwritten digit dataset was classified by a neural network, achieving the lowest error rate to date [221]. Success continued with AlexNet, a classifier trained on 1.2 million images was proposed by Krizhevsky et al. [106]. This classifier was the first of its kind with such a large training set. The proposed approach utilised a CNN which encompassed 60 million parameters and 650,000 neurons and predicted 1000 classes of image. The dataset in question is known as ImageNet and provided a platform for researchers to compete in order to achieve the best classifier [52]. AlexNet indicated a new wave of neural network based deep learning models. Krizhevsky et al. found their approach to provide considerably better results than the previous state-of-the-art for the same dataset [106]. ImageNet lay the ground work for a raft of future convolutional neural network (CNN) based image classifiers besting previous methods by statistic significance, including the creation of VGG [190], and others [237, 183, 189] in 2014. Later in 2016 [82] pushed the neural network boundaries further, exploring deeper networks. He et al. found that *very deep* networks suffered from vanishing gradients. Gradients between network nodes become increasing small as the network deepens with network weights having increasingly small effects and subsequent learning becoming stationary. He et al. found that forcing propagation (through skip connections) of the higher layer features allowed this problem to be overcome and gradients to be maintained in much deeper structures [82]. He et al. coined the architecture name ResNet, standing for residual network [82]. ResNet development continued with depths ranging from 18 layers to 152 layers. Success was seen with ResNet on large scale problems such as ImageNet [82], beating AlexNet in 2015.

Time series algorithms are a different type of classifier that have previously been developed and studied to solve a range of real world problems. Sempena et al. used Dynamic Time Warping (DTW) to recognise human body positions and actions from imagery depth maps [182]. Sempena et al. found that DTW proved effective at recognising human body shapes when comparing video capture against pre-defined actions [182]. Deecke et al. also used DTW to categorise cetacean tonal sounds [**Deecke**]. In 2014 Chen et al.[35] used a Bayesian network to accurately classify species of flying insects. Chen used low cost sensors in order to record laser fluctuations onto a digital sound recorder. Chen found using a Bayes classifier provided accurate classification results with minimal CPU and memory requirements, whilst also being easy to implement and having no parameters to tune [35].

A broad range of detection and classification methods have been applied more specifically to cetacean detection in recent years. Edge and threshold detectors have been used to detect odontocete whistles [70] and NARW upcalls [69]. Time series methods such as vector quantisation and dynamic time warping have been effective in detecting blue and fin whales from their frequency contours extracted from spectrograms [148]. Hidden Markov models (HMMs) have also been effective at recognising low frequency whale sounds using spectrogram features [140]. Deep learning investigations comparing artificial neural networks (ANNs) and spectrogram correlation for NARW detection [139] have also been made. Further to the use of ANNs, support vector machines (SVMs) have been applied effectively to odontocete classification [96]. SVMs have been compared against Gaussian mixture models for classification of three types of odontocetes [169]. Classifying NARWs with convolution neural networks (CNNs) has also been investigated, however using Mel-frequency coefficient input features instead of standard spectral representations [191]. More recent work in 2020 conducted by Shiu et al. [188] provides a comprehensive review of neural network techniques for NARW detection, finding neural networks to be effective at detection with an large increase in accuracy compared to methods proposed in a 2013 conference challenge. Recent trends and success in the field of machine learning have influenced the subsequent techniques used to detect NARWs within this work. This work considers techniques that represent the acoustic recordings in different forms to understand how best to detect NARWs.

3.3 Development of deep learning algorithms

This section aims to develop and test a range of deep learning approaches for detecting NARW vocalisations from acoustic recordings. In recent years, deep learning has seen large improvements in areas where time series traditionally returned the best results [106, 59]. Computer vision and speech recognition have seen the greatest advancement from the rise of deep learning algorithms [220] with some applications outperforming human performance for the same task [121, 233]. Three types of neural networks will now be developed; fully connected networks (FCNs), recurrent neural networks (RNNs) and convolutional neural networks (CNNs). As previously discussed in Section 3.2, FCNs, RNNs and CNNs form the basis for deep learning as they are known today. FCNs are the simplest and therefore offer the smallest processing requirements. RNNs have been specifically developed for tasks that benefit from knowledge of past data. As the problem of NARW classification relies on important temporal information, RNNs should provide valuable recurrent properties to facilitate classification. CNNs are known for their excellent performance in image classification applications. CNNs provide an alternate transformation of the initial time domain data. Comparison of these approaches will be invaluable in evaluating the best method for NARW classification.

The reminder of Section 3.3 is as follows. Section 3.3.1 investigates feature extraction parameters for NARW vocalisations. Section 3.3.2 explores both time domain and spectral domain input features, as well as considering a range of network architectures for fully connected neural networks. Section 3.3.3 similarly investigates time domain and spectral domain input features but instead develops a recurrent neural network for classification. Finally, classification from a CNN is then explored in Section 3.3.4, where a range of feature extraction parameters are examined to provide the most suitable features for classification.

3.3.1 Feature extraction for North Atlantic right whale vocalisations

The purpose of feature extraction is to transform the input audio signal into a representation that is more effective for detecting whale sounds. Although many different methods of audio feature extraction have been developed (for example Mel-frequency cepstral coefficients [MFCCs], perceptual linear prediction [PLP] and filterbank [143]) a standard power spectral representation was chosen in this work. MFCCs are a method of representing acoustic frequencies in such a way that aims to mimic the human auditory system [143]. Frequency bands are equally spaced, however they are on the Mel-scale [50] instead of being linearly spaced as found on the normal spectrum.

Feature extraction using MFCCs was designed to produce features that more closely map human speech and are therefore more suitable for speech recognition tasks. PLP approach feature extraction similarly [84], designed to produce features most suited to human speech. Considering the original purpose of these techniques and due to the unknown differences between the human and cetacean auditory systems these methods were not investigated further. Using the standard power spectral representation therefore allows the subsequent networks (FCN, RNN or CNN) to learn discriminative representations and not remove what could be useful information, such as may happen when using, for example, a mel-scaled filterbank.

Input features are often dependent on the chosen machine learning technique. Methods aim to use the most appropriate input features for the chosen classifier. For example the time series methods all initially use the time domain audio signal as they have been developed to work best on a time series. Neural networks however are targeted as generic algorithms aimed at accepting a wide range of input features. It therefore is crucial to investigate the most suitable input features whilst also assessing other parameters such as the network architecture. Extraction of the power spectrum is now explored with further parameter testing in Section 3.3.1.

Power spectrum

The process of feature extraction to create power spectral features (often referred to as spectrograms, when stacked temporally) uses a sliding window to convert short-duration frames of the input audio signal into a sequence of log power spectral vectors, $\mathbf{x}_t$. Specifically, an N-point frame of time-domain samples is extracted from the audio, Hamming windowed and a Fourier transform computed. The upper $N/2$ frequency points are discarded and the remaining points logged. An overview of the process

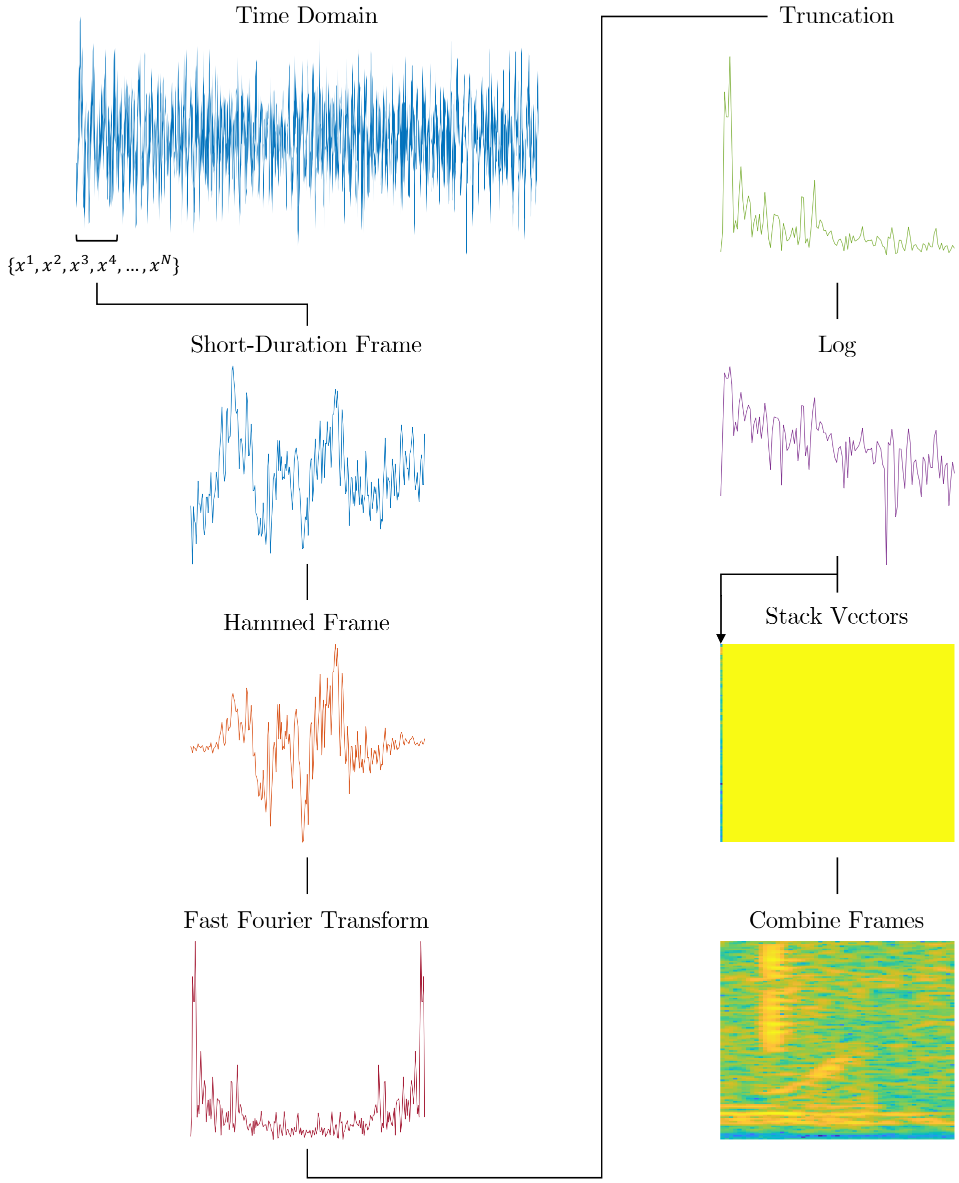


Fig. 3.1 Overview of the stages involved when creating a spectrogram comprising a series of power spectral vectors taken from an audio source. This diagram shows the processing of one power spectral vector. This process occurs across the entire signal to produce the final spectrogram seen in the bottom right.

is detailed in Figure 3.1. Analysis windows are advanced by S samples to compute each new spectral vector. At a sampling frequency of f_s Hz, a total of $\frac{f_s-N+1}{S}$ spectral vectors are computed each second. This gives the total number of time-frequency points, L, that are produced each second as

$$L = \frac{f_s - N + 1}{S} \times \frac{N}{2} \tag{3.1}$$

Normalisation is applied to the elements of the power spectral vectors such that they are in the range 0 to 1. The power spectrum offers an enhanced insight into the frequency of recordings as time progresses. Frequency contours from marine mammals are often clearly visible within the power spectrum and as such provide a good starting point for classification rather than from the time domain signal. Figure 3.2 shows a comparison of the same signal in both the time domain and the spectral domain. In Figure 3.2 the NARW upcall is clearly visible in the spectral domain (right), whereas the upcall is not visually present within the time domain (left). In Figure 3.2 the spectrogram (right) has been extracted with a sampling frequency of 1kHz, and a 32ms time resolution. Typically for automatic speech recognition a frame width of 10-30ms and a 50% frame overlap is common. This is based on the human autonomy of vocal organs, however it is understood that similar parameters may not be optimal for NARW vocalisations.

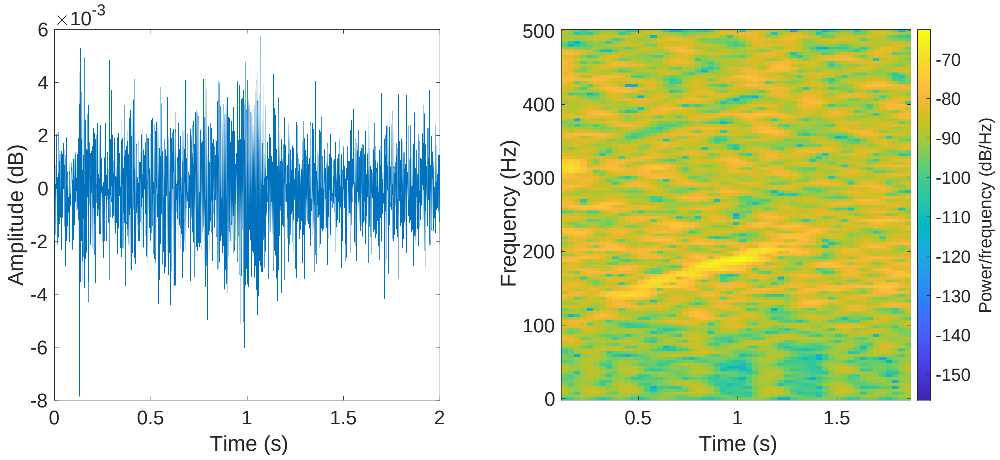


Fig. 3.2 Comparison of a single audio file shown in the time domain (left) and spectral domain (right).

Spectrogram parameters

NARW classification in the spectral domain is based on first extracting a time-frequency spectral feature from the audio signal and inputting this into a classifier to predict the presence of a NARW. The time-frequency feature, $\mathbf{X}$ is created using the process detailed in Section 3.3.1. Within $\mathbf{X}$ each element x_{ij}, represents the energy at time index i and frequency index j. During the creation of spectral features, N, the window size, S, the slide of the window and f_s, the sampling frequency, are all parameters that can dramatically alter the generated feature. The first parameter, N, effects the frequency resolution. Choosing a smaller window length will produce the effect seen in Figure 3.3 where frequency resolution decreases as N becomes smaller and the frame is made up of more spectral vectors.

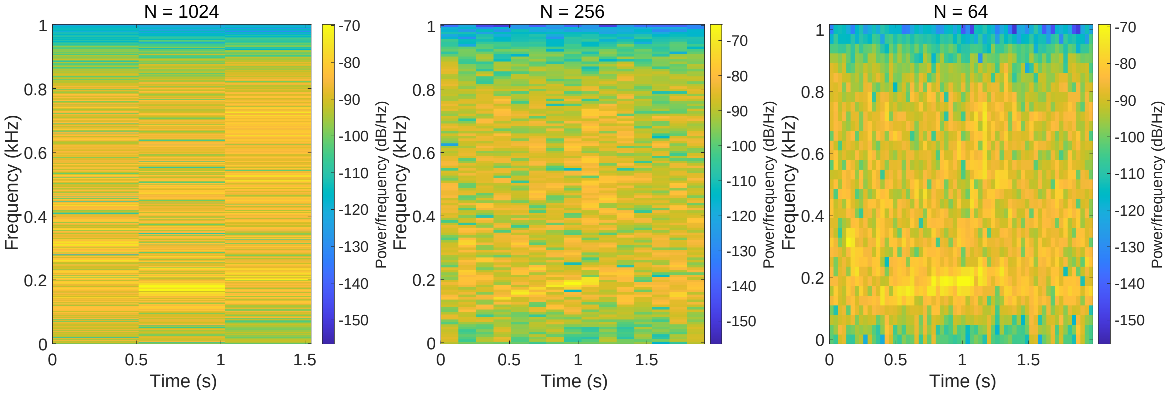


Fig. 3.3 Spectrograms with their N value set to 1024, 256 and 64 respectively. Other parameters are $S = N$ with $f_s = 2000Hz$.

The second parameter, S, accounts for the slide of each window. Having a smaller S means a larger amount of short-duration frames are captured with each frame overlapping the previous. The spectrograms in Figure 3.4 use a fixed $N = 256$, with $S = 256, 128, 1$. These S values represent a non-overlapping window, a half overlapping window and a window that advances by only a single frame, causing 255 overlapping frames. The change of S seen in Figure 3.4 shows the clarity change when using overlapping frames, capturing a greater level of temporal detail. Computing more frames does however lead to increased computation. Therefore a balance between, S and accuracy can often be found. An increase in S eventually saturates and does not directly correlate to clearer spectrograms, as can be seen in images 2 & 3 in Figure 3.4.

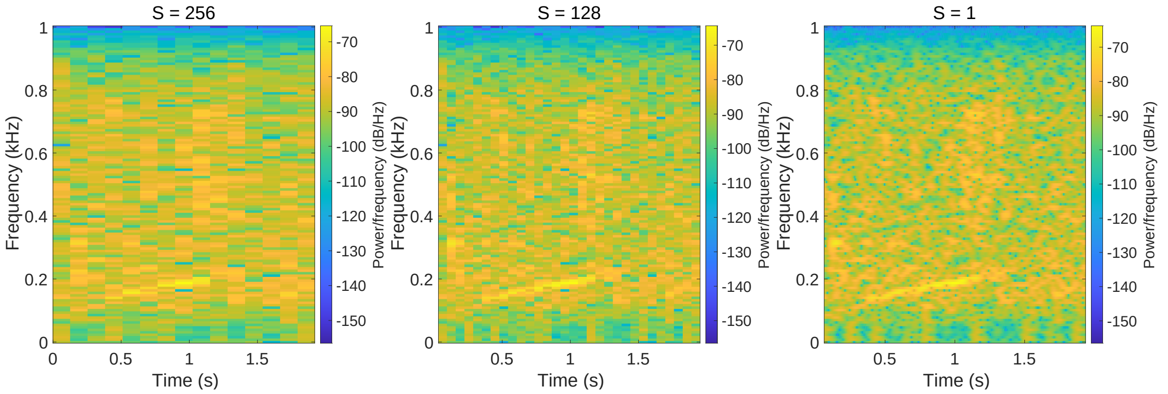


Fig. 3.4 Spectrograms with their S value set to 256, 128 and 1 respectively. Other parameters are $N = 256$ with $f_s = 2000Hz$.

The third parameter, f_s, is the sampling frequency of the audio. Using a smaller f_s requires the original data to first be resampled. Figure 3.5 uses the original f_s, resampled to a half, and resampled to a quarter. Both $N = 256$ and $S = 32$ were fixed for consistency, however N, S and f_s have a relational link. They all effect the number of short-duration frames captured when f_s is reduced. The effect seen in Figure 3.5 is similar to cropping, with higher frequencies discarded. Testing a reduction in f_s indicates the optimum frequency capture range for NARWs but also can provide potentially faster computation as the lower frequency features will have smaller dimensions. All plots in Figure 3.5 are stretched to fit the given area, however it should be noted that the frequency axis (left) is reducing by a factor of 2 on each subsequent plot. Tests in Section 3.3.4 investigate the most suitable spectrogram parameters for NARW vocalisation classification.

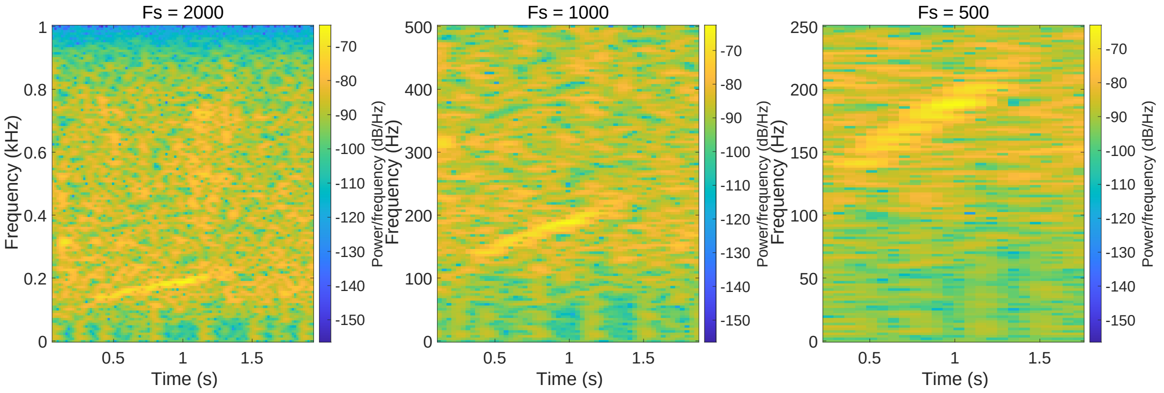


Fig. 3.5 Spectrograms with their f_s value set to 2000Hz, 1000Hz and 500Hz respectively. Other parameters are $N = 256$ with $S = 32$.

3.3.2 Fully connected networks

Fully connected networks (FCNs) are often referred to by their simpler counterpart, artificial neural networks (ANNs) or deep neural networks (DNN) however deep networks could refer to any neural structure with more than one hidden layer. In the context of this thesis FCNs will describe a neural network structure with an input layer, output layer and one or more dense layers. This structure will reflect many network types but FCNs specifically only utilise dense (hidden or fully connected) layers between the input and output layers.

FCN structure

Figure 3.6 provides a visual description of a FCN with two dense layers, fed by an input layer and returning an output layer (class prediction). A network of this structure can take input from any source, however the input would need to be first transformed into vector. For example, an acoustic source is already in the correct form as a one dimensional signal, however an image such as a spectrogram would first need conversion into a vector prior to processing. For a spectrogram this could be achieved by stacking frequency vectors into one sequence.

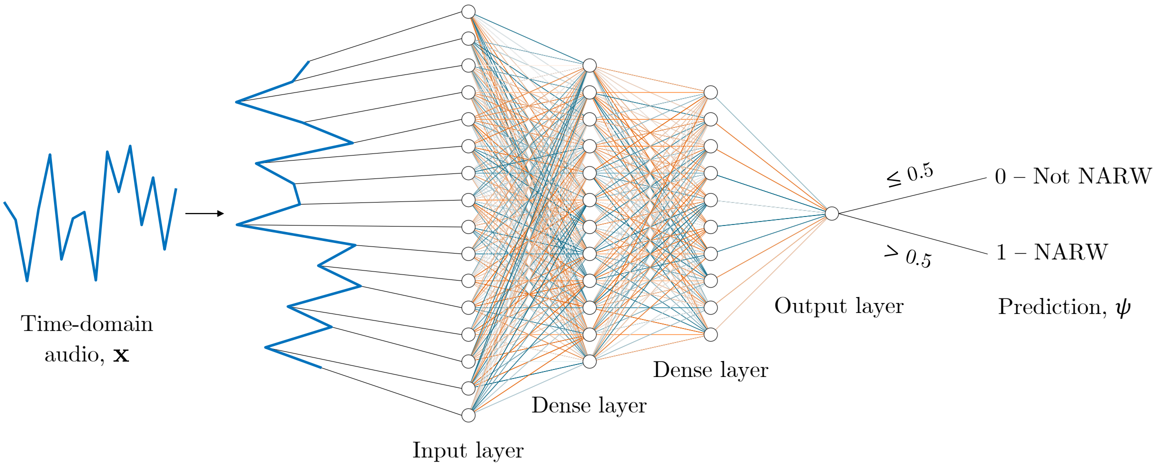


Fig. 3.6 Visualisation of a fully connected network fed from an acoustic signal, $\mathbf{x}$. The network structure contains an input layer connected two dense layers and a single output node for binary classification.

FCNs are made up from a number of fundamental mathematical operations that act on a combination of the input data and network parameters. FCNs, compared to other network types, use the simplest set of operations, with each dense layer taking input from all nodes on the previous layer. Neural networks contain a vast amount of user-defined parameters and functions. Understanding the relationship between these enables a deeper level of exploration when maximising classification accuracy for NARW classification. A FCN is defined by a number of attributes, such as the architecture (number of layers and nodes), training process, and optimiser.

1. **Architectural elements**

A number of different variables contribute to the fully connected network architecture. These are the layer type and number of nodes. FCNs comprise three layer types, input, dense and output layers with each responsible for a specific operation.

(a) **Input layer**

The input layer $x^{[1]}$ defines the size and shape of the network input. Input throughout the network will be noted as $\mathbf{x}$ with $x^{[1]}$ specifically referring to the input of the first level - the input layer. The input data must conform to the input layer shape otherwise a mapping from the data to input shape must be defined. The input layer can take a range of forms for different types of input data [38] or different neural network architectures, however for an FCN the input will present as a vector. As an example, in Figure 3.6 the input layer is a $(1, 16)$ vector representing the time domain signal of an acoustic recording taken from the ocean.

(b) **Dense layer**

Dense layers are inward facing layers and cannot take input or generate output directly. Dense layers are often referred to as hidden or fully connected layers however, for consistency, will be referred to as dense layers. Dense layers take input from all nodes on the previous layer to produce an output. This flow of data creates a many-to-many relationship between all nodes on dense layers within the network [166]. This can be see on Figure 3.6 between the first dense layer and the input, and for a single node in Figure 3.7.

When designing a network architecture, both the number of dense layers and number of nodes per layer need to be chosen. Using a large number of nodes and layers will enable greater granularity of information within the network. However computation time and size will increase and excess complexity may be introduced into the model. The term ‘node’ refers to a single position within a layer. Dense layers are formed from a set of nodes, as can be seen in Figure 3.6. At each node, a combination of the output from the previous layer, $x_i^{[l-1]}$, connection weights, $w_i^{[l]}$, and network bias, $b_i^{[l]}$ are computed to produce the pre-activation output, $z_i^{[l]}$, shown in Equation 3.2.

(c) **Output layer**

The output, ψ normally presents as a layer with the number of nodes matching the number of classes for that classification task. Each node represents a class of the data with the output of all nodes, after application of a sigmoid function, totalling one. The highest valued node signifies the class that the model predicts the input data to belong to [38]. For binary problems a network may use a single node with an output value $\psi > 0.5$ belonging to class 1 and $\psi \leq 0.5$ to class 2. Initial NARW tests are presented as a binary problem as shown in Figure 3.6. Accuracy of the model is measured by comparing the model prediction, ψ to the label, g. The percentage of correctly classified inputs produces the model accuracy.

2. **Node components**

Nodes sit within each layer of the network and are used to process each data point. Each node has a number of components that contribute to its activation.

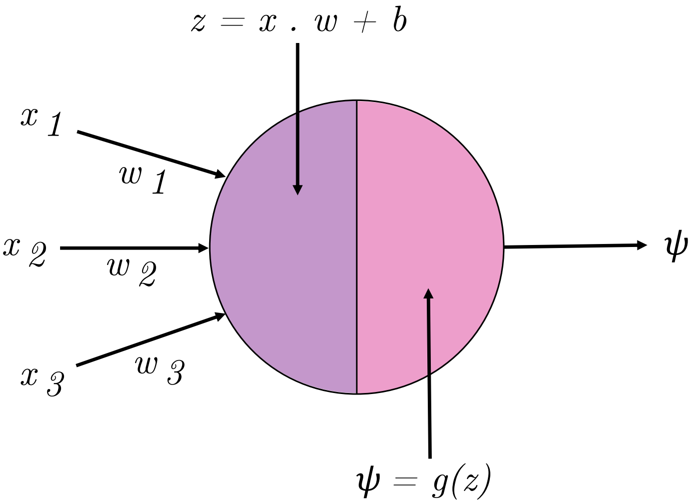


Fig. 3.7 A diagram to represent the flow of data within a dense node.

$$z_i^{[l]} = \sum_{i=1}^{N^{[l-1]}} (x_i^{[l-1]} \cdot w_i^{[l]}) + b_i^{[l]} \tag{3.2}$$

(a) **Weights**

Connection weights, $w_i^{[l]}$ are present on every internal node connection between the input and output layers. Weights define the amount of movement each node can have when training the model. During each training pass, the model weights can be updated and subsequently alter the output value of each node.

(b) **Bias**

A bias, $b_i^{[l]}$ is a constant that is added to the output of each node and ensures the desired node output is reached to activate each node. This value is again updated through the training process.

(c) **Activation**

The activation of a node defines the final node output before 'firing' to the next node. Attaching a function to manipulate the output, enables more complex patterns within the data to be learnt as non-linearity can be introduced. For problems where the classes do not separate linearly, a linear output such as $z_i^{[l]}$ may struggle to differentiate the classes effectively. Providing the introduction of a non-linearity has shown to be highly effective at enabling the network to learn more complex structures [124].

For the models in this work, in general, Rectified Linear Units (ReLU) are used [149]. Activation of a node is shown in Equation 3.3 with the ReLU function applied to $z_i^{[l]}$ before being passed onto a subsequent layer. ReLU is one of many non-linear activation functions but is used here as its use within the field of deep learning is wide spread, with recognition of its ability when dealing with complex models [152].

$$x_i^{[l]} = g(z_i^{[l]}) = max(0, z_i^{[l]}) \tag{3.3}$$

3. **Loss function**

The loss function provides a method of assessing model performance for each piece of input data. A loss value, $\mathcal{L}(\psi, g)$ is the result of the loss function aiming to provide an accurate measurement between the class label, g and model prediction, ψ. In this work, for two class problems, binary cross-entropy was used as the function to generate $\mathcal{L}$ [43]. The aim of the network is to minimise $\mathcal{L}$, indicating that the set of w is producing ψ that more closely match g. Equation 3.4 describes binary cross-entropy where N is the number of classes - for NARW classification this is two. For ψ values distant from their g counterpart, Equation 3.4 will generate a larger $\mathcal{L}$ value, which signifies a larger w shift is necessary in the subsequent update of the model parameters.

$$\mathcal{L}(\psi, g) = -\frac{1}{N}\sum_{i}^{N}[g_i log(\psi_i) + (1 - g_i) log(1 - \psi_i)] \tag{3.4}$$

4. **Training and Optimisation**

Training is the process of allowing the network to *learn*. A defined number of network passes are allocated for the model to learn from the training data - a subset of the entire dataset. A further subset is reserved for validating the model's performance during training, with a third split separated for testing. Test data is never processed by the model during training and is therefore entirely unseen; this reflects the real world situation of processing new acoustic recordings. During training, each piece of training data will pass through the network, producing a prediction, ψ. In order for the network to learn and improve, an optimiser is used in conjunction with $\mathcal{L}$ during backpropagation. The network will be trained for a limited number of epochs and the weights from the best performing model will be used to label new instances of ocean recordings.

An optimiser aims to change the connection weights, w, to reduce loss and to produce a higher number of correct classifications, which for NARW detection would be more NARW vocalisations correctly identified. Different optimisers are available to use, however throughout testing, Adam [103] was used as it consistently achieves fast convergence and excels at finding the global minimum $\mathcal{L}$ [153]. Adam calculates the gradient between each weight, $w_i^{[l]}$ and $\mathcal{L}$ by utilising the partial derivative of $\mathcal{L}$ with respect to $w_i^{[l]}$. Equation 3.5 examines an overview of the Adam algorithm. The result of Equation 3.5, $w_i^{[l]}$, provides an updated weight to the network. In Equation 3.5, α defines the learning rate. The learning rate can be updated manually or automatically based on the learning speed of the network. Controlling the learning rate allows for larger or smaller weight changes during backpropagation.

$$w_i^{[l]} = w_i^{[l]} - \alpha \frac{\partial \mathcal{L}}{\partial w_i^{[l]}} \tag{3.5}$$

Backpropagation is the process of updating network weights during training [75]. Every time a piece of training data passes through the network every activation function, on each individual node fires, passing values onto the subsequent layer and finally producing a network prediction, $\hat{y}$. Each time this occurs an updated $\mathcal{L}$ is generated. Backpropagation occurs by traversing backwards through the network updating each connection with an updated $w_i^{[l]}$ calculated using the optimiser. Once the entire network is updated the next forward pass can occur with a new piece of training data. This circular loop enables the learning process to take place.

Every time a full pass of the training data occurs, the model is said to have completed an epoch of training. Deciding on the number of epochs for training is an important decision when designing a neural network. Using a low number will mean training takes less time but potentially means the model will not have reached a global minima $\mathcal{L}$ and given more time to train, could become more accurate. Alternatively a high number of epochs will ensure the model is sufficiently trained with convergence on a global minima $\mathcal{L}$ highly probable, however wasted computation may occur if the model converges before the end of training.

Time domain testing

The initial experiments into NARW classification utilised the time-domain audio signal for input in the FCN and use the Cornell corpus as defined in Chapter 2.6.3. Since the time-domain signal occupies a vector, no manipulation was necessary to process the raw audio using the FCN. A number of network parameters are available to alter during testing. Prior to further investigation into each hyper-parameter, the core parameters; number of dense layers, and number of nodes per layer, are investigated. Investigating these parameters provides analysis of how the different architectures perform and subsequently informs if further analysis of network hyper-parameters is necessary. In order to extensively evaluate performance of the FCN, a range of layers and nodes were tested. Table 3.1 details an overview of tests, with the number of layers ranging from 1 to 40 and nodes per layer ranging from 2 to 256. A large range of architectures were evaluated to enable clear understanding of how the network was performing and which architecture provided the best results. The best performing architecture is highlighted in bold. All tests were repeated 10 times as the network starting weights are randomised for each test. An average of all 10 repetitions is calculated and presented.

Table 3.1 demonstrates the necessity for FCN architectures to use a sufficient number of nodes on each layer. Near best network performance can be achieved with only 10 layers and 128 nodes per layer. A further gain in performance can be found when increasing the number of layers, with the maximum performance found when using 25 layers and 128 nodes per layer. Convergence appears to occur at this point with further tests showing no improvement in accuracy for the increase in model complexity.

		Nodes per layer							
		2	4	8	16	32	64	128	256
Network depth (layers)	1	54.17	54.78	51.22	51.73	52.01	51.41	50.97	51.07
	5	51.46	57.21	65.08	62.95	62.05	61.25	50.30	50.06
	10	50.53	60.51	67.00	62.98	67.93	68.96	69.19	67.67
	15	50.00	55.58	67.89	68.53	69.83	69.69	69.33	70.38
	20	50.00	55.35	66.27	69.47	69.73	69.59	70.06	70.08
	25	50.00	52.70	67.43	69.52	70.20	70.17	**70.85**	70.65
	30	50.00	51.67	62.85	68.95	69.27	69.88	70.31	69.68
	35	50.00	50.00	57.56	69.80	69.67	69.22	69.92	68.39
	40	50.00	51.72	58.67	69.13	70.00	70.23	69.79	69.17

Table 3.1 Accuracies achieved from a wide range of network architectures for FCN. Network depths range from 1 to 40 dense layers with nodes per layer ranging from 2 to 256.

It is suggested that 128 nodes performs well because this provides enough compression of the original audio without removing an excess of data points that are necessary to differentiate the classes. It is also thought that a minimum number of 10 layers is necessary to accurately model the complexity of the raw audio. Tests with fewer layers struggled significantly.

Table 3.1 indicates limited performance using a FCN network on the time domain signal. Previous research in the field of signal processing indicates that the raw audio signal is unsuitable for speech recognition [113, 99] and potentially therefore poor for use as an input feature in a FCN for NARW classification. Since the time domain signal does not include any frequency information, it is understandable that the FCN performance had a ceiling of 70.85% accuracy. These results provide a platform to evaluate alternative deep learning methods against.

Spectral domain testing

As previous research concludes [113, 99, 87], power spectral features are effective at providing frequency analysis of an audio source. Representing this as a spectrogram also allows for temporal information to be included and as such could be important for making classifications. Further tests therefore explore the use of spectrogram features (Section 3.3.1) for input into the FCN. To match the correct input dimensions the spectrogram must first be transformed into a vector for computation. Parameters for spectrogram creation were taken from later work in Section 3.3.4, where a 1kHz sampling frequency, 3.9Hz frequency resolution and 32ms time resolution performed best. This spectral resolution gives spectrogram parameters of $N = 256$ and $S = 32$ and produces a matrix of 129 × 55 time-frequency points and subsequently produces a 7,095 point vector when transformed. The same testing framework as first seen in Section 3.3.2 was used to analyse performance of spectral features.

Table 3.2 shows the accuracy when testing across a range of network depths and nodes per layer. The highest accuracy is shown in bold. Using the spectral domain enables far greater accuracies when compared to the time domain with a 16.8% improvement between the maximum for both methods. Results in Table 3.2 indicate that spectral features contain more insightful class information as the model can predict more correct labels, over results for time domain tests in Table 3.1. However, Table 3.2 shows that the architecture is more volatile with many variations unable to produce a detection rate greater than chance (i.e. 50% for a 2 class problem). Best performance is found when using 15 layers and 16 nodes per layer, however using a range of 32 to 128 nodes and 2 to 15 layers produced extremely similar accuracies and shallower networks with less nodes are computationally cheaper, an important requirement for real-time classification. Since the optimal network for spectral domain features is significantly smaller in both layers and nodes per layer than that of the best performing time domain network it is thought that the spectrogram features contain a larger amount

		Nodes per layer							
		2	4	8	16	32	64	128	256
Network depth (layers)	1	50.00	50.00	53.65	57.48	53.79	72.08	57.50	50.00
	2	50.00	53.74	57.41	80.01	87.55	87.50	87.40	87.41
	3	50.00	55.61	68.75	80.18	87.52	87.54	87.36	87.40
	4	50.00	57.48	61.27	80.18	87.36	87.18	87.49	87.44
	5	50.00	57.48	68.76	80.17	87.41	87.29	87.33	87.30
	10	50.00	53.74	80.17	83.71	87.39	87.25	87.11	87.26
	15	50.00	53.70	72.48	**87.65**	87.25	87.40	87.33	75.03
	20	50.00	50.00	72.55	87.31	87.46	87.18	82.52	53.77
	25	50.00	50.00	50.00	68.78	84.07	68.77	57.52	50.00
	30	50.00	50.00	50.00	53.71	53.75	53.68	50.00	50.00
	35	50.00	50.00	50.00	53.75	53.75	50.00	50.00	50.00
	40	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00

Table 3.2 Accuracies achieved from a wide range of network architectures with flattened spectrograms used as input into a FCN. Network depths range from 1 to 40 dense layers with nodes per layer ranging from 2 to 256.

of discriminative information and therefore requires less nodes and network depth to differentiate between classes.

3.3.3 Recurrent neural networks

Recurrent neural networks (RNNs) have recently found widespread use in many modern applications, such as; natural language processing (NLP) [224], language translation [37], and speech recognition [77] that have temporal structure. The idea of recurrent networks is not new with the initial concept discovered in 1980 [50], however only since the early 2010s have the real advantages to them been unveiled with the growth of big data [20] and faster GPU-accelerated computing [187]. The benefit of recurrent networks for specific applications is that they account for temporal information within the input. Tasks such as speech recognition therefore greatly benefit from the context of previously classified speech before making predictions of whole words or sentences. Taking into account the benefit of RNNs, it was logical that time-frequency feature classification could benefit from the temporal learning properties of the network over that of the static modelling with FCNs [213].

RNN structure

A characteristic of FCNs is that they have no memory. Each input shown is processed independently, with no state kept in between inputs [38]. To fully understand streams of information where context is embedded in previous sections, it is essential that a memory of past events is kept. RNNs adopt this principle by processing sequences of information whilst concurrently maintaining a state relative to what has been seen so far [38]. A feedback loop connected to each node allows this to occur, by feeding seen information back into the current node.

The structure of an RNN comprises of a sequence of recurrent layers, each with a number of nodes, followed, optionally, by dense layers. A limitation of RNNs is the diminishing gradients problem [92]. Diminishing gradients occur when the network weights are updated by small gradients that cause insignificant change for that node. In the worse case, training would stop as gradients become too small to make any change. Gradients become small as certain activation functions reduce large input spaces into small ones, often in the $0 - 1$ range. The derivative of these outputs, used to calculate the gradient, becomes even smaller, in time producing diminishing gradients [91]. To avoid diminishing gradients each RNN layer is implemented using a long short term memory (LSTM) cell [90]. All LSTM layers use the hyperbolic tangent activation function with dense layers using a ReLU activation. LSTM layers enable data seen previously to be forcefully propagated into deeper nodes [90]. This propagation helps to increase gradients as the *memory* of an LSTM cell is not subject to processing from an activation function and larger input spaces can stay large. LSTM layers will now be explored in detail to show the information flows through a LSTM cell.

1. **LSTM layer**

LSTM layers are a variation on original recurrent layers [175] as they enforce a continuous memory property, which is a stream of data that passes chosen information further down the network. This *'memory cell'*, C_i allows each node to see previously processed data as well as upcoming data. LSTM layers have been widely adopted in place of recurrent layers as they can achieve higher accuracy [186] due to their ability to gain discriminative features from temporal information. Therefore only LSTM layers are used when testing RNN configurations. LSTM nodes contain three main decisions gates; input, forget and output. Each gate is responsible for including or disregarding information from the input data. Figure 3.8 shows a diagram of an LSTM node. All inputs, outputs and gates are shown. The internal memory cell, C_i provides a long term memory for the network and enables later nodes to see data from previous timesteps. Each gate works to add or remove information from C_i in order to propagate relevant features further into the network. C_i thus reduces the effects of short-term memory when compared with traditional recurrent nodes. Each gate contains a sigmoid function to reduce values to between 0 and 1 for easier propagation or removal. For simplicity network weights, w_i^g, are not included in the equations below or on Figure 3.8. Each gate within the node has a set of weights, w_i^g, with g defining the gate. Weights work as they did previously for FCNs with values updated during training and backpropagation in order to shift importance of information being passed through the node.

(a) **Forget gate**

The forget gate is responsible for removal of non-relevant information from the memory. By removing this information a larger proportion of class defining values are kept and classification becomes easier. Firstly the input, X_i, is concatenated with the

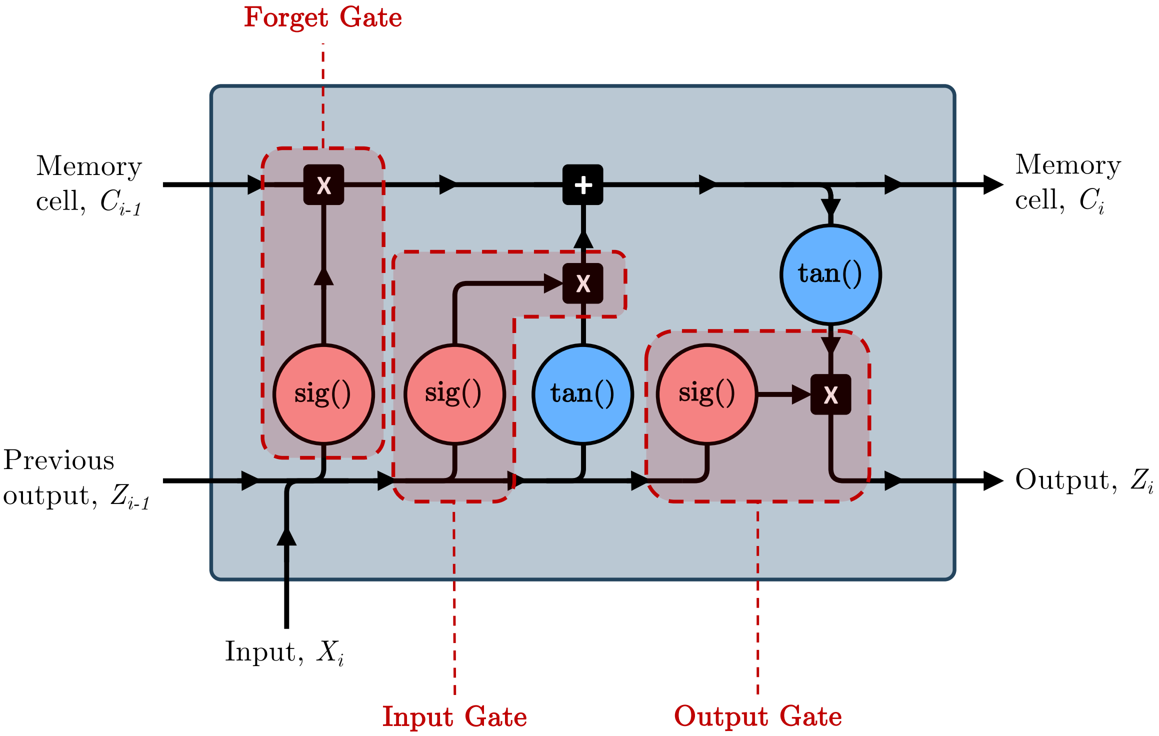


Fig. 3.8 A diagram of a LSTM module showing the input, output and internal functions and also detailing the three module gates.

previous output, Z_{i-1} and passed through a sigmoid function shown in Equation 3.6. Overtime, the network learns to minimise unwanted values and maximise important values. Equation 3.7 shows the output of the first sigmoid, f_i is then combined with the C_{i-1} to remove values that were minimised. Concatenation of two vectors is represented with a $\frown$ in the upcoming equations.

$$f_i = \sigma([X_i \frown Z_{i-1}]) \tag{3.6}$$

$$C_i = C_{i-1} \cdot f_i \tag{3.7}$$

(b) **Input gate**

The input gate defines values that improve performance during training. The input gate specifically aims to maximise values which help to separate classes for classification. The input gate passes a concatenation of the input, X_i and previous output, Z_{i-1} through a sigmoid function, producing j_i, shown in Equation 3.8. The same values are also passed through a tanh function to produce a candidate memory cell, $\tilde{C}_i$ (Equation 3.9), which combined with j_i, signals which values within $\tilde{C}_i$ to maximise. $\tilde{C}_i$ is then added to C_i to push the new values into memory, shown in Equation 3.10.

$$j_i = \sigma([X_i \frown Z_{i-1}]) \tag{3.8}$$

$$\tilde{C}_i = tanh([X_i \frown Z_{i-1}]) \tag{3.9}$$

$$C_i = \tilde{C}_i \cdot j_i \tag{3.10}$$

(c) **Output gate**

Similarly to the other gates, the output gate applies a sigmoid function to input, X_i and previous output, Z_{i-1} to produce o_i in Equation 3.11. The output gate decides, based on the memory cell, what the output of the current node should be. After producing o_i the memory cell has a tanh function applied, before being multiplied by o_i and producing the output, Z_i, which is detailed in Equation 3.12. Both Z_i and C_i are passed onto the next node.

$$o_i = \sigma([X_i \frown Z_{i-1}]) \tag{3.11}$$

$$Z_i = tanh(C_i) \cdot o_i \tag{3.12}$$

		Nodes per layer							
		4	8	16	32	64	128	256	512
LSTM depth (layers)	1	80.25	87.41	89.31	90.30	90.52	90.50	90.49	90.35
	2	86.55	88.54	89.51	90.43	90.43	90.42	90.37	90.42
	3	88.20	88.96	89.71	90.50	90.53	90.50	90.29	84.60
	4	88.69	89.37	89.96	90.59	**90.78**	90.61	90.59	65.24
	5	89.10	89.69	89.92	90.57	90.61	90.62	90.11	50.00

Table 3.3 Tests evaluating a range of LSTM layers and nodes for a RNN classifier.

RNN architecture tests

The same input feature previously used in Section 3.3.2 where $N = 256$, $S = 32$ and $f_s = 1000Hz$ is used to test RNN architectures for NARW classification. Testing for the CNN in Section 3.3.4 found a multi-layered encoder and 2 layer classifier to achieve the best results. Tests in this section keep the classifier consistent with Section 3.3.4, whilst manipulating the number of LSTM layers and nodes per layer. Table 3.3 shows the accuracies reported from these tests. A range of high performing architectures were found using 1 to 5 layers and between 32 and 128 nodes. The best accuracy was produced with 4 layers and 64 nodes, with other shallower networks not being significantly different. This shows that deeper models have less effect than the number of nodes per layer. It is thought that shallower models could perform well due to the use of the LSTM cells and their propagating memory. Deeper models are unnecessary as the latent space can be smaller due to memory cells containing crucial discriminative information. Through all tests, except those with 512 nodes, a deeper model provided a minimal gain in accuracy and at times reduced performance. A compromise in the real world to maximise performance whilst minimised processing requirements would be to use a network containing a single LSTM layer with 64 nodes, combined with 2 dense layers.

3.3.4 Convolutional neural networks

Convolutional neural networks (CNNs) have recently become popular due to their ability to successfully classify large datasets of images [106] such as Imagenet, a collection of images containing over 1000 classes [52]. CNNs work similarly to FCNs, utilising a layered structure to process inputs, however convolutional layers are applied prior to dense layers and these extract features from the input before dense layers produce a final prediction. Since 2012, image recognition research has been centered around deep learning and specifically CNNs [190, 189, 82]. It is therefore important to investigate whether CNNs can be applicable to NARW vocalisation recognition by treating spectrograms of PAM recordings as images.

CNN structure

Figure 3.9 shows a block CNN structure for classifying NARW with convolutional layers placed before dense layers. The whole structure in Figure 3.9 makes up a CNN however the convolutional layers specifically make up an encoding block as they learn to extract data specific features from the input spectrogram to create a compressed representation. For example, masking background noise within a NARW classification and only keeping the pixels that pertain to this frequency contour. As an example, Figure 3.10 shows a comparison of a typical input spectrogram against the output of a 3rd layer convolutional filter. As Figure 3.10b shows, the network has propagated the upcall pixels through the layers with the rest of the image becoming unstructured noise. Extracting data specific feature enables the dense layers to work more effectively at separating classes for classification.

Many functions within a CNN are the same as their FCN counterpart. For example, activations, backpropagation, optimisers, epochs and loss functions. Further

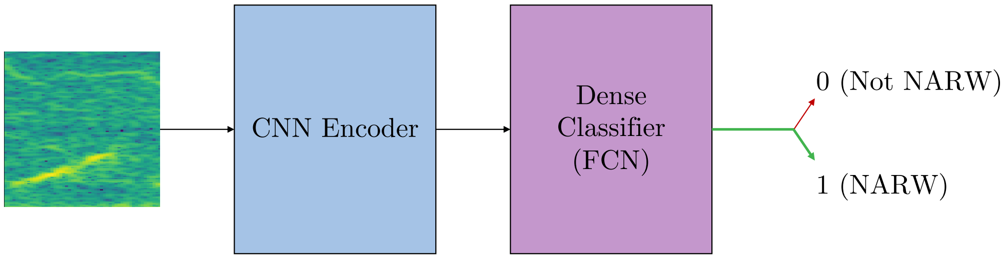


Fig. 3.9 A standard CNN structure, defining the position of convolutional layers in relation to the dense layers.

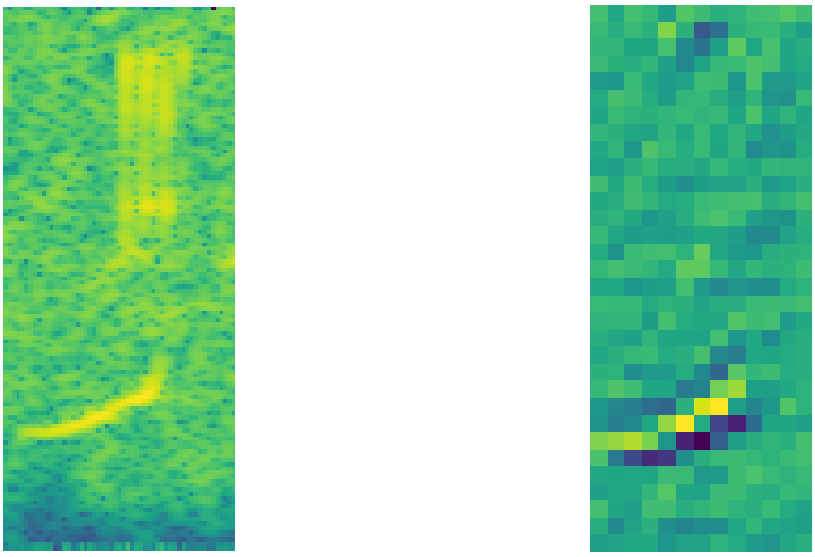
explanation of CNN specific operations will now be discussed.

1. **Convolutional layer**

Convolutional layers aim to find features within the input spectrogram, $\mathbf{X}$, and propagate these deeper into the network. Features found in images are normally details or edges that define the image class. For example the feature that defines a NARW upcall is the upsweep in frequency producing a contour seen on Figure 3.10a.

(a) **Filters**

Filters (or kernels), $\mathbf{W}$ are generally small patches of randomly initialised weights [38] that move over the entire input, $\mathbf{X}$. The aim of using filters is to encode specific aspects of the input data, such as edges or lines within the image, helping to separate that image from others [38]. Filter sizes are user defined but are often 3×3 or 5×5. For specific applications, any filter size can be used, such as 1×5 or 5×1 to capture horizontal or vertical detail. Often multiple filters are applied within a single convolutional layer. Using multiple filters has the advantage of enabling a range of filter designs to be applied and produces a broad output of feature maps to be propagated through the network, however too many filters can cause unnecessary complexity for an unfounded gain in network performance. Previously, $w_i^{[l]}$ defined the weights on



(a) Input feature

(b) Activation output

Fig. 3.10 A comparison of the an input spectrogram against the output of a 3rd convolutional layer filter operation within a CNN.

connections between nodes within a FCN. For CNNs, **W** refers to convolutional filter weights and these are learnt and updated during backpropagation. Updating **W** allows the network to stabilise the best performing filters for use on unseen data. Well trained filters can extract the most appropriate features for classification.

(b) Convolution operation

Each convolutional layer is designed to produce the product of the input and each filter, in order to generate a feature map, **S**, related to that filter. The convolution operation convolves **W** over **X** until every pixel has been seen. Each filter operation produces the product of the filter, **W** and the pixels of the input covered by the filter. Equation 3.13 shows this sliding window approach, producing an output feature map of $S(f,h)$, where f and h represent the output feature map dimensions.

$$S(f,h) = \sum_{a}\sum_{b} W(a,b) \cdot X(f+a,h+b]) \qquad \text{for all f and h} \tag{3.13}$$

Due to the convolution process, **S** is smaller in both the i^{th} and j^{th} directions compared to **X**, with the reduction in size controlled by the size of the filter. To counteract this, padding can be used to add a single border of zeros, giving **X** an artificially larger size. When using padding the subsequent feature map matches the dimensions of the given input.

(c) Pooling

Pooling is an operation when using a convolutional layer that reduces dimensionality by pooling together values from a larger input. Multiple types of pooling exist however commonly used methods are; *max* and *average.* Both max and average-pooling operate in the same manner with max taking the maximum value of a pool, whereas average-pooling, takes the average value. Pooling aims to reduce dimensionality whilst maintaining previously extracted features such as edges within the image. Pooling windows are usually small and often 2×2. The pooling process is applied in a similar way to the convolutional operation, and uses a sliding window across **S**. Unlike the convolutional operation, each pool does not overlap but instead moves across **S** seeing each value only once. The pooling window moves across **S** and extracts either the maximum or average value from the window. The output then creates a smaller feature map **P**, usually half the size of **S**.

CNN input feature

In order to evaluate a range of input features, a baseline CNN setup was used, with alternate architectures investigated in Section 3.3.4. The CNN consisted of the network shown in Figure 3.11 and used a binary cross entropy loss and Adam optimiser. The batch size was 128 with the model trained for 100 epochs. As previously stated, each test was repeated 10 times with the average accuracy reported.

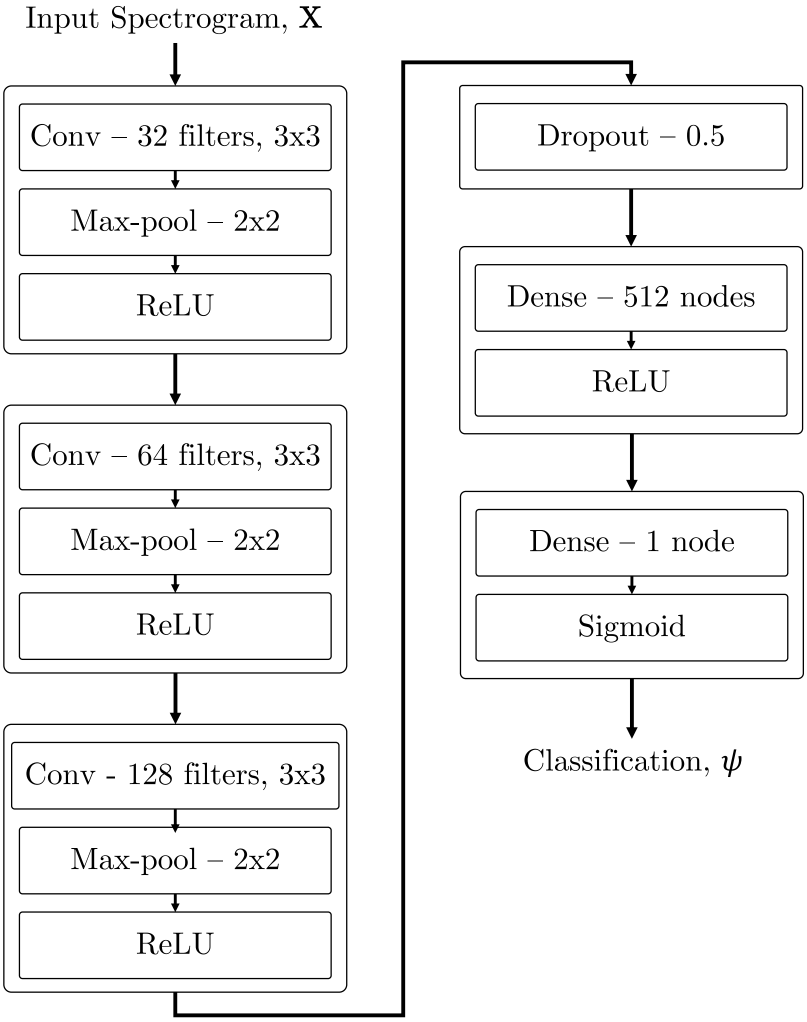


Fig. 3.11 CNN architecture consisting of three convolutional layers with max-pool and ReLU functions, followed by two dense layers, the final used to output the model prediction.

Tests now examine the trade-off between accuracy and processing time by examining the time and frequency resolution of the input feature. Frame widths between 256ms and 16ms are considered first with a fixed 50% overlap of frames which gives a time resolution, Δt, between 128ms and 8ms. In terms of the frequency resolution, Δf, this varies between 3.9Hz and 62.5Hz, depending on the window size and sampling frequency.

	Δt	128ms	64ms	32ms	16ms	8ms
2kHz	Δf	3.9Hz	7.8Hz	15.6Hz	31.3Hz	62.5Hz
2kHz	L	3584	3840	4032	3968	3984
2kHz	Accuracy	91.4%	**92.1%**	91.6%	90.2%	89.9%
1kHz	Δf	3.9Hz	7.8Hz	15.6Hz	31.3Hz	62.5 Hz
1kHz	L	1792	1920	1952	1984	1992
1kHz	Accuracy	91.2%	**92.0%**	91.6%	90.6%	90.0%

Table 3.4 Classification accuracy and number of points for varying time and frequency resolution features with 50% frame overlap.

The number of time-frequency points, L, for each configuration is computed using Equation 3.1. For each time resolution, Table 3.4 shows the resulting frequency resolution, Δf, number of time-frequency points, L, and classification accuracy, for sampling frequencies of 2kHz and 1kHz. A 500Hz sampling frequency was also evaluated, however accuracy degraded, therefore 500Hz was not included in further testing [214]. Highest accuracy for both sampling frequencies is found with the 64ms-7.8Hz time-frequency resolution, with 92.1% for 2kHz and 92.0% for 1kHz sampling frequencies. Considering the number of points, and hence processing time, the 1kHz system requires half the computations and gives almost equal performance to the 2kHz system.

The tests in Table 3.4 were performed with 50% frame overlap which means that frequency resolution deteriorates as time resolution improves. The parameters are now considered independently by allowing the frame overlap, S, to vary while keeping the frame width fixed. Specifically, two fixed frame widths are considered to give high and low frequency resolutions of Δf={3.9Hz, 15.6Hz} and the frame slide adjusted to give varying time resolutions, Δt, from 64ms to 8ms. The resulting accuracy and number of time-frequency points are shown in Table 3.5 for 2kHz and 1kHz sampling frequencies.

For both frequency resolutions and both sampling frequencies the time resolution has relatively little effect between 64ms and 16ms, with highest accuracy at 32ms.

	Δt	64ms	32ms	16ms	8ms
2kHz	Δf	15.6Hz	15.6Hz	15.6Hz	15.6Hz
2kHz	L	1984	3904	7808	15552
2kHz	Accuracy	91.1%	**91.6%**	91.0%	90.0%
2kHz	Δf	3.9Hz	3.9Hz	3.9Hz	-
2kHz	L	7168	14080	28160	-
2kHz	Accuracy	92.1%	**92.3%**	91.3%	-
1kHz	Δf	15.6Hz	15.6Hz	15.6Hz	15.6Hz
1kHz	L	992	1952	3904	7776
1kHz	Accuracy	91.0%	**91.6%**	91.5%	91.0%
1kHz	Δf	3.9Hz	3.9Hz	3.9Hz	3.9Hz
1kHz	L	3584	7040	14080	28032
1kHz	Accuracy	92.3%	**92.5%**	91.6%	91.0%

Table 3.5 Classification accuracy and number of points for varying time resolutions against frequency resolutions of 15.6Hz and 3.9Hz.

In terms of frequency resolution, the finer resolution gives higher accuracy across all configurations tested, although this comes at the cost of increased processing time. For example, highest performance of 92.5%, with 1kHz sampling frequency, 3.9Hz frequency resolution and 32ms time resolution used 7,040 points. This could be reduced to 1,952 points (corresponding to a processing time three times faster) by using a wider frequency resolution but with a reduction in accuracy to 91.6%.

CNN architecture tests

NARW classification via a CNN is based on first extracting a time-frequency spectral feature from the audio signal and then inputting this into a CNN to predict the presence of a NARW. Spectrogram parameters taken from the previous section provide the most suitable input features, and tests are now focused on finding the best performing CNN architecture. To create the input features, parameters of $f_s = 1000Hz$, $N = 256$, $S = 32$ were used. In order to compare architectures, available network parameters were separated and tested independently to monitor their influence. Tests were carried out in two stages. First, convolution depth and number of filters per convolutional layer were evaluated. Second, tests evaluating the number of dense classification layers and nodes per layer were performed, similarly to the tests on the FCN in Section 3.3.2.

When first investigating a suitable number of convolutional layers and filters, the number of dense layers was fixed at two. The first dense layer contained 512 nodes and the final layer had 1 node and was used to output the classification. Tests explored network depths of 1 to 5 with deeper models not possible due to the reduction in feature size caused by the max-pooling operations. Table 3.6 shows the accuracy for a range of convolutional layer depths and filters per layer. Results in Table 3.6 show that a network with 3 convolutional layers and 64 filters per layer achieved the highest accuracy, with 32 and 128 filters also achieving similar accuracies. Peak performance is seen using 3 convolutional layers with worse accuracy when using fewer layers and similar accuracies but with extra computation for deeper networks.

		Filters per layer					
		4	8	16	32	64	128
Convolutional depth (layers)	1	83.69	71.97	71.97	86.35	50.00	50.00
	2	90.14	90.21	90.41	90.49	90.28	90.35
	3	90.91	91.20	91.64	91.78	**91.87**	91.79
	4	90.60	91.29	91.54	91.52	91.78	91.74
	5	89.20	91.31	91.50	91.23	91.32	91.43

Table 3.6 Tests evaluating a range of convolutional layers and filters for a CNN encoder.

Further testing fixed the convolutional layers to the optimum found in Table 3.6 (3 layers with 64 filters) and varied the number of dense layers and nodes per layers. The highest accuracy was achieved with 2 dense layers and 256 nodes per layer. The results in Table 3.7 shows that only a few dense layers following convolutional layers are required to achieve the highest accuracies. On many tests, a depth of 2 dense layers

		Nodes per layer							
		4	8	16	32	64	128	256	512
Dense depth (layers)	1	87.22	91.46	91.47	91.50	91.41	91.61	91.51	91.87
	2	87.26	91.35	91.60	91.53	91.44	91.58	**92.03**	91.64
	3	91.46	87.12	91.32	91.62	91.42	91.41	91.46	91.39
	4	87.22	87.35	91.36	91.32	91.39	91.57	91.29	91.37
	5	83.15	87.30	91.35	91.40	91.38	91.47	91.61	91.29
	10	74.78	87.25	91.38	91.31	91.42	91.52	91.46	88.75
	15	58.35	70.81	87.10	91.22	91.34	87.30	66.62	54.06
	20	50.00	50.00	66.59	78.92	78.97	54.15	50.80	50.00

Table 3.7 Tests evaluating a range of dense layers and nodes for a CNN classifier.

provided the highest accuracy for the number of nodes.

Summary of deep learning methods

Thorough network testing discovered the highest performing model achieved 92.03% accuracy however earlier tests, involving the input features, saw a slightly varied architecture achieve 92.50%. The initial architecture utilised a tiered number filters, progressing from 32 to 128 over three convolutional layers, as shown in Figure 3.11. Additional tests found a model combining these two architectures to achieve the highest performance seen on this dataset. An accuracy of 92.62% was found using three convolutional layers, each followed by a max pooling and ReLU function. In all convolutional layers, 3×3 filters were applied with padding at the edges, with 32, 64 and 128 filters in each layer respectively. A dropout of 0.5 was used with two dense layers to form the classifier. The dense layers contained 200 and 50 nodes respectively each with a ReLU activation function. A final output layer uses a single node with a sigmoid function to provide predictions. Figure 3.12 shows a visualisation of this CNN architecture. This model is marked as the proposed CNN for NARW classification and is used to carry out all further CNN tests. This network is later compared in Section 3.5 against the other ML methods to assess which technique is best overall for NARW classification.

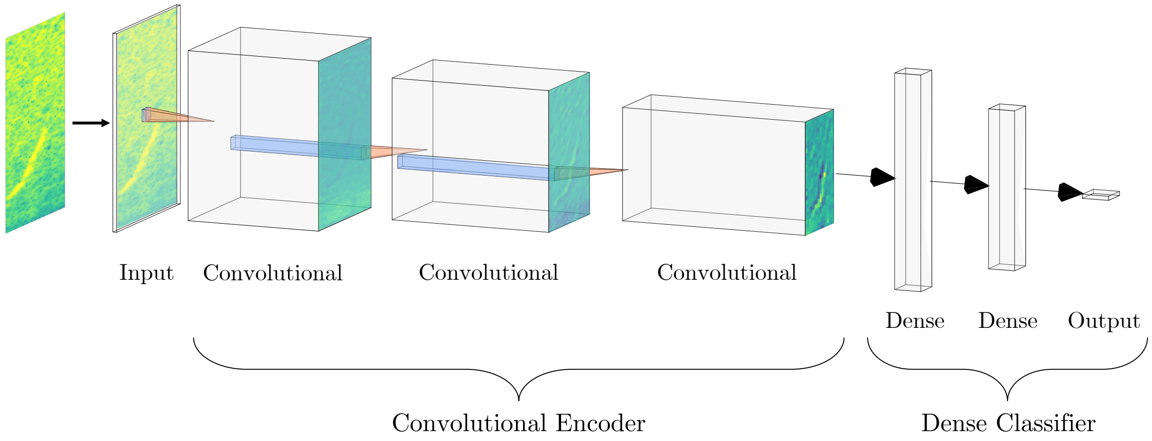


Fig. 3.12 A diagram to show to the best performing CNN architecture after testing with activation outputs superimposed onto their location within the network.

3.4 Time series algorithms

The aim of this section is to present a range of time series methods that have been applied to NARW classification. Time series methods will be compared and their performance, when detecting NARW upcalls will be analysed. A traditional approach to audio classification is to use the audio signal directly to form a time series classification (TSC) problem. The vast majority of TSC algorithms operate on time domain data as, until recently, the consensus was that a '*simple nearest neighbour classification is very difficult to beat*' [12]. As such, much emphasis has been placed on developing approaches for solving problems in the time domain using alternative elastic distance measures with nearest neighbour classifiers [195, 101, 129]. However, not all TSC algorithms actually operate on the time domain data when classifying. Therefore, for evaluation purposes, two subsets of TSC algorithms are considered; time domain methods and feature domain methods. Time domain methods utilise the time series throughout execution whereas feature based methods apply a transform to the time domain data before execution.

3.4.1 Time domain methods

The most popular TSC time domain approach is dynamic time warping (DTW) with a warping window set through cross-validation and 1-nearest neighbour (DTW 1NN). While it has been shown that ensembling different elastic nearest neighbour classifiers can be significantly more accurate [116], combining such *lazy* classifiers increases test classification run-time. With a large amount of training data, necessary for capturing the range of NARW signals and background noises, classification processing times are likely prohibitive for real-time deployment on an ASV, so for this application DTW 1NN is used as a benchmark for the time domain approaches.

3.4.2 Feature domain methods

Feature domain methods operate by transforming the time domain signals into an alternative representation where discriminatory information is more easily detected. A recent comparison of approaches [8] demonstrated that best performance is obtained by combining the output of a range of classifiers built over various representations of a problem to produce combined predictions from a meta-ensemble [117]. However, given the processing limitations in implementing classification on ASVs, this would not be practical but suitable constituent transformation-based approaches may produce fast, accurate results. In particular, three such constituents are considered: 1) *time series forest* which is built on summary features from phase-dependent intervals [51]; 2) *shapelet transform* which is a heterogeneous ensemble using data transformed by similarity to phase-independent discriminatory subsequences [86]; 3) *RISE*, random interval spectral ensemble which is a forest-based ensemble classifier that builds constituents using features extracted from the auto-correlation and power spectral domains [117].

3.4.3 Comparison of time series algorithms

The aim of these experiments is to explore the accuracy of the time series classification methods and to consider these in respect to the trade-off against processing requirements. The methods compared are, i) dynamic time warping (DTW), ii) time series forest (TSF), iii) shapelet transform (ST) and iv) random interval spectral ensemble (RISE). DTW provides representation for time domain methods, with TSF, ST and RISE all operating in the feature domain and covering a range of approaches for feature transformation algorithms. Figure 3.13 shows classification accuracies for all four TSC methods across three sampling frequencies; 500Hz, 1000Hz and 2000Hz to investigate the effect that reducing sampling frequency has on classification accuracy. DTW, the only time domain classifier achieves the lowest accuracy, with performance consistent across all sampling frequencies. In all cases, the remaining methods achieve a minimum of a 10% increase in accuracy over DTW with performance increasing by as much as 22% in the best case using shapelet transform at 500Hz. ST achieves the highest accuracy overall with RISE and TSF falling in accuracy.

Tests were conducted over a range of sampling frequencies in order to ascertain if a reduction in data points would decrease accuracy. All feature domain methods saw an increase in accuracy when using 500Hz recordings. The NARW upcall signal of interest commonly lies within the 50Hz-250Hz frequency range however can occasionally appear higher in the frequency spectrum. It is therefore logical that classifier performance improves with downsampled audio as this removes unnecessary frequencies before classification and enables a potentially cleaner, more specific model to be generated.

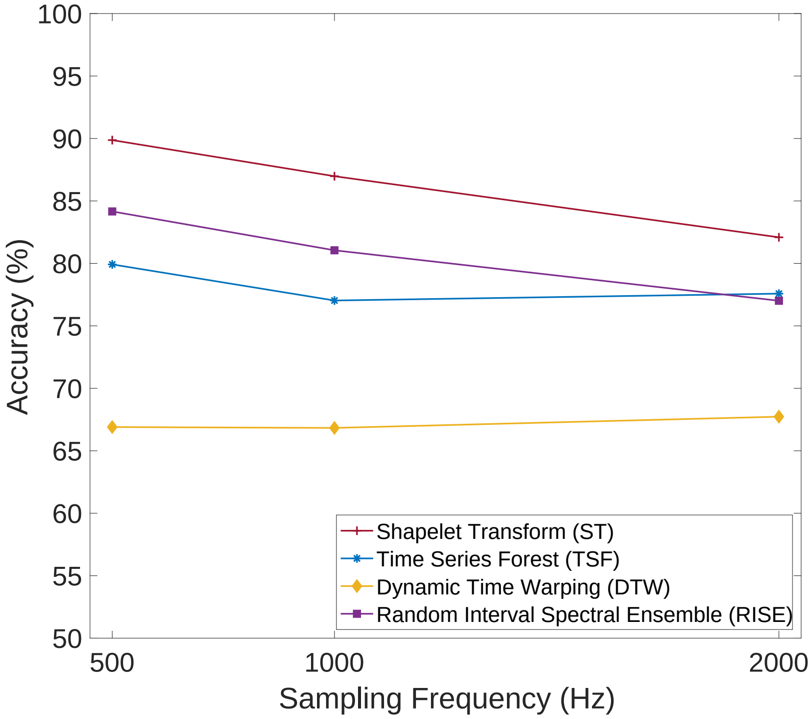


Fig. 3.13 Comparison of classification accuracies provided by four machine learning algorithms tested on three sampling frequencies from 500Hz - 2000Hz.

The additional benefits of downsampling are two-fold; firstly, downsampling recordings compresses the size of data files and hence provides storage benefits when detecting via an autonomous surface vehicle, reducing necessary physical storage by up to 75%; secondly, processing requirements are reduced as the signal is downsampled, this allows for cheaper and less power-intensive hardware to classify the audio in real-time.

3.5 Comparison of machine learning methods

In this section, the FCN, RNN and CNN are compared to investigate which performs best when detecting NARWs. Other networks such as VGG [190] and ResNet [82] are well documented and widely used a baseline for image classification performance.

These networks contain more complex architectures with specific properties, such as residual learning, and are targeted at large-scale image classification problems such as the original ImageNet competition [176]. VGG and ResNet are therefore also evaluated within this work. These network architectures will be referred to as, 'pre-made', as they have already been tested and tweaked for maximum performance. They are used here as a benchmark for the current state-of-the-art methods. The pre-made networks operate with starting weights defined by the ImageNet dataset. Each model was trained for a further 100 epochs from the starting weights to give a final classification accuracy. Shapelet transform (ST) is also compared as this is the highest performing time series method from Section 3.4.3.

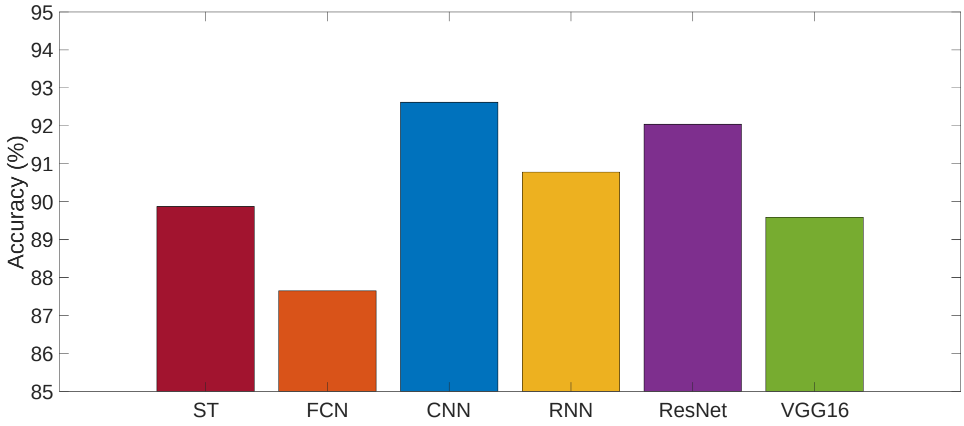


Fig. 3.14 Comparison of accuracies for all machine learning methods under test.

Figure 3.14 shows all deep learning networks under test with the best performing architectures from Sections 3.3.2, 3.3.3 & 3.3.4. For consistency all networks used the spectrograms parameters discovered in Section 3.3.1 as input and trained for 100 epochs. Each experiment was run 10 times and results shown are the average over the 10 runs. Figure 3.14 shows the CNN achieved the highest accuracy of 92.6% when compared to the other methods with ResNet closely behind achieving 92.04%.

3.6 Discussion

A range of deep learning and time-series methods have been applied to the classification of NARWs with best performance, in terms of both accuracy and processing time, given by the CNN. Downsampling the audio leaves accuracy almost unchanged but gives a substantial reduction in processing time which is advantageous for processing onboard an ASV. Considering time and frequency resolutions reveals that a wide resolution of 32ms gives good accuracy whilst higher frequency resolutions are better, albeit at increased processing cost. Many of the classification methods provide high levels of accuracy with deep learning methods generally performing best. Pre-made networks attained good results but were far more complex using 50 layers than the shallower CNN architecture using 3 layers. Since the problem of NARW classification only contains two classes it is suspected that deeper models are unnecessarily complex (and computationally expensive) for the required binary class mappings.

Chapter 4

Investigation and application of conventional methods of audio and image based noise reduction

4.1 Introduction

This chapter investigates both audio and image based conventional methods of noise reduction for improving NARW detection in noisy conditions. Noise reduction is aimed at reducing background noise within a signal to enhance the NARW vocalisation and ultimately make detection more accurate. In this work, the methods presented are applied to PAM recordings to enhance the NARW vocalisation for subsequent classification using a CNN, as this was judged to be the most effective method in Chapter 3.

Tests within this chapter aim to improve system robustness and explore the effectiveness of detection in more noisy conditions. As noise conditions change, a mismatch between a clean-trained model and data collected in the new noise condition is introduced; this mismatch ultimately leads to a fall in detection accuracy. Performance on the system can be improved by training a new model to match to the new conditions, but this is time consuming and is not always practical in real world situations as conditions change. For example, if a clean-trained model was being used to detect NARW whilst at sea on-board a ship, a new noise condition could cause a dramatic reduction in detection accuracy. Retraining a model in this scenario would be impractical as training a model cannot be carried out in real-time and collecting data takes time to label accurately before use by the model.

Using noise reduction or an enhancement technique is a potential way of recovering performance when the audio contains high levels of background noise. This chapter investigates a range of traditional enhancement techniques to improve the accuracy of the proposed CNN, developed in Chapter 3.3.4, when new noise conditions are present. These can be considered as two types of approaches, i) those using traditional enhancement applied to the audio signal, and ii) those using image enhancement applied to spectrogram features. The methods investigated have previously been successful in audio and image based applications and consequently form a good starting point for this work.

First, Wiener filtering, log spectral amplitude (LSA) estimation and spectral subtraction (SS) methods are applied to the original time domain audio. Audio-based enhancement methods are used here as they are traditionally found in speech enhancement [22] and work effectively to reduce noise and enhance speech [1, 54]. The principle is similar when applied to the NARW recordings, however the signal of interest is different to that of human speech.

The second set of techniques are point processing and histogram equalisation methods, which are applied to spectrogram image representations of the audio. Image-based enhancements can be used as the spectrogram features they are applied to can be treated as images and present visual details of the recorded frequency content. In Chapter 3.3.4 it was discovered that using spectrograms to represent the time-frequency content of each audio recording worked effectively to detect NARW vocalisations. Image-based enhancement therefore targets the spectrogram representation to enhance vocalisations and suppress background noise. Point processing techniques have previously been successfully used to enhance images for greater human and computer recognition [125]. Histogram enhancement methods are also effective in the field of computer vision and image processing with methods being successfully applied to many tasks such as improving image dynamic range in dark conditions [118, 232] and improving contrast in medical imaging applications to better assess patients [160, 244].

The remainder of this chapter is organised as follows. Section 4.2 gives a in-depth background into previous methods of audio and image enhancement, specifically exploring noise reduction for cetacean detection. Audio-based enhancement methods are then developed in Section 4.3 and their effectiveness when applied to noisy NARW recordings is investigated. Similarly, image-based enhancements of NARW recordings are investigated in Section 4.4 with their effectiveness reported. Initial tests examine the performance of each enhancement technique when using a classifier trained on noise free data. Subsequent tests examine performance on a model retrained with an enhanced corpus.

4.2 Background

Noise reduction is a crucial step in many modern audio processing systems in order to produce a clean and non-corrupt signal. Noise reduction helps to reduce background noise for tasks such as; speech recognition [89, 120], audio playback [142] and video conferencing [83, 155, 198]. These applications all benefit from having noise reduction applied, either by an increase in performance or improved user experience. Investigations into the reduction of noise for detection of NARW is however less common, with automated detection of marine mammals only gaining popularity more recently [97, 188, 71, 14]. Since anthropogenic ocean noise is known to cause to marine mammal injury and death [25, 55], with prolonged exposure to loud noises increasing the risk of incident greatly [64], it is of paramount importance to make accurate detections in areas of noise. Shipping lanes, ports, oil/gas rigs and wind farm construction sites all generate substantial noise [223, 167] and are areas where mitigation zones for marine mammals are in place in order to protect their populations [3].

In audio enhancement applications, an input signal, $\mathbf{y}$ often contains an element of noise, $\mathbf{d}$ such that

$$\mathbf{y} = \mathbf{x} + \mathbf{d} \tag{4.1}$$

where $\mathbf{x}$ is the clean signal [122], which in this work is a NARW vocalisation. Previous work [122] defining noise reduction techniques, reduces the output signal, $\mathbf{y}$, by a noise estimate, $\widehat{\mathbf{d}}$ to give a estimated clean signal, $\widehat{\mathbf{x}}$, where

$$\widehat{\mathbf{x}} = \mathbf{y} - \widehat{\mathbf{d}} \tag{4.2}$$

The three audio-based enhancement methods evaluated in Section 4.3 are spectral subtraction [122], Wiener filtering [122], and log spectral amplitude (LSA) [122]. The objective of these methods is to produce an estimation of the clean signal. This is shown in Figure 4.1 with the audio enhancement method producing a noise-free estimate that is subsequently converted to a spectrogram before detection via the proposed CNN system. Yi Hu and Philipos Loizou [94] carried out an extensive comparison of speech enhancement techniques, specifically investigating which methods subjectively produce signals that sound best after processing. These tests evaluated methods in noise conditions using SNRs of 0dB, 5dB, 10dB and 15dB [94]. Hu and Loizou found that both Wiener and LSA methods performed well against others within their categories with LSA performing the best across all noise conditions [94]. Spectral subtraction did not perform as well as LSA or Wiener [94], however due to its popularity for audio enhancement it is used in this work as a baseline comparison with the other methods. Although this chapter considers NARW detection, work based upon speech enhancement operates within the same domain and should therefore be considered when analysing techniques to reduce noise. Speech enhancement methods aim to enhance a signal within noisy audio, this is aligned with the requirements of a robust NARW detector, aiming to enhance NARW vocalisations.

In Section 4.4, a range of image-based enhancement methods are investigated. These are split into two groups, point processing methods in Section 4.4.1 and histogram equalisation methods in Section 4.4.2. Both sets of methods are applied in the spectral domain by considering each spectrogram of audio as an image. As an example, Figure 4.2 shows the process of enhancing the spectrogram of an upcall using an image enhancement prior to detection via the proposed CNN system. Point processing methods operate on each point of the input feature, directly relating to

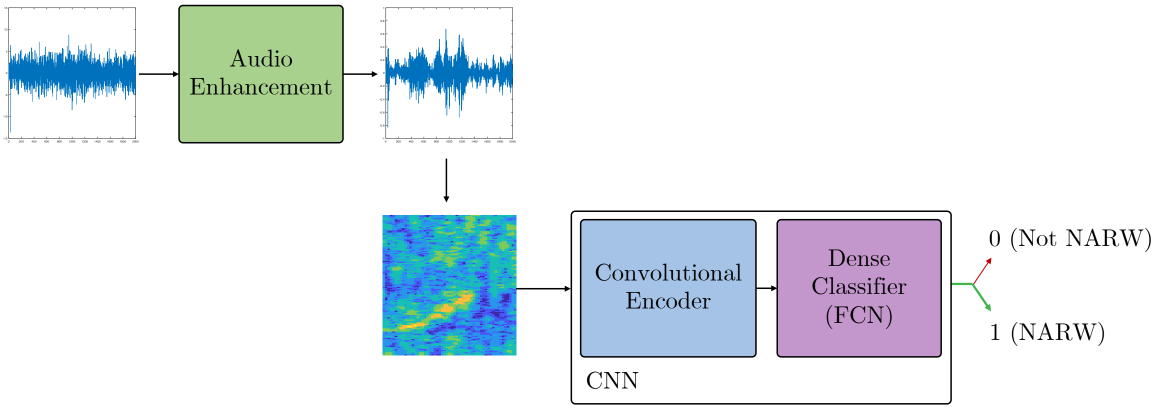


Fig. 4.1 Diagram to represent the process of audio enhancement prior to detection via the baseline CNN system. Wiener filtering is used to create the example shown.

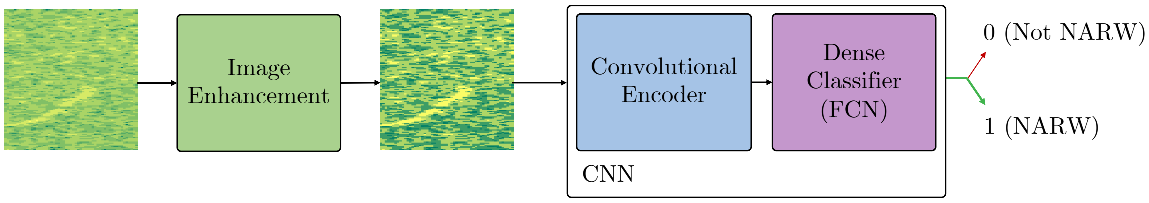


Fig. 4.2 Diagram to represent the process of image enhancement prior to detection via the baseline CNN system. Histogram equalisation is used to create the example shown.

a subsequent point of the output. Maini [125] carried out a review of point-based image enhancement techniques and found although many are primitive operations, they are fundamental in providing image enhancements, often providing richer contrast or exposing details in dark or bright regions; which for the spectrograms used in this work are regions of high or low energy. A range of point processing techniques are evaluated in Section 4.4.1 to compare their ability of providing enhancements in harsh conditions, where low energy vocalisations can be masked by high energy background noise.

Histogram equalisation (HE) is a method applied to the colour spectrum of an image to equalise the intensity values [74]. HE aims to flatten a histogram of colour intensities to achieve equal intensity across the image [74]. This has the effect of brightening regions of darkness whilst darkening regions of brighter light. HE helps to evenly distribute colour within the image and often reveals details, otherwise unseen. This process can often improve visual details in harsh conditions, such as high levels of noise [74]. Although on an audio application, Schuller et al. [179] used histogram equalisation to improvement contrast in MFCC feature vectors for speech recognition in noisy conditions. Schuller et al. found that HE could improve speech recognition accuracy by adding to the robustness of their system when recognising speech in noisy conditions [179].

Further work in the histogram equalisation domain by Zuiderveld [245] established a technique known as contrast limited adaptive histogram equalisation (CLAHE) which develops upon HE with contrast equalisation occurring over regions (tiles) within an image. Each tile is individually equalised leading to a higher level of contrast control unlike HE which is performed across the entire image [245]. Further work analysing human retinas saw CLAHE to dramatically improved visual image detail when examining retinal scans [184]. Kumar et al. also saw large improvements in SNR when using CLAHE against other contrast-based image enhancement methods for underwater imagery [108].

4.3 Audio enhancement methods

Three audio-based enhancement methods are now considered for enhancement of NARW vocalisations. All methods are applied to the time-domain signal and output an enhanced signal. The aim of these methods is to suppress background noise and enhance the signal of interest, in this case, the NARW vocalisation.

4.3.1 Spectral subtraction

Spectral subtraction (SS) is the most simple method of noise reduction considered, as it only requires a noise spectral estimate [80]. SS can however introduce residual noise, leaving artefacts after noise removal [26]. SS first transforms the noisy signal, $y(n)$, into the frequency domain and extracts a magnitude or power spectral representation, $|Y(f)|^2$. An estimate of the noise magnitude spectrum, using a short duration frame from the being of the audio, $|\hat{D}(f)|^2$, is then subtracted to give an estimate of the clean magnitude spectrum, $|\hat{X}(f)|^2$, [107]

$$|\hat{X}(f)|^2 = |\hat{Y}(f)|^2 - |\hat{D}(f)|^2 \tag{4.3}$$

After subtraction, the enhanced power spectrum is combined with the noisy phases and an inverse Fourier transform applied to return the signal back into the time-domain. Its effectiveness relies on the accuracy of the noise estimate, and the assumption that noise is relatively stationary between update periods, and its performance can therefore vary. Good results have been found in stationary environments, with SS struggling in more practical non-stationary settings [107].

4.3.2 Wiener filtering

The Wiener filter has a more involved implementation than SS and requires not just a estimate of the contaminating noise, but an estimate of the SNR at each frequency bin [80]. The gain of the Wiener filter, $W(f)$, at a frequency, f, is given as

$$W(f) = \frac{|X(f)|^2}{|X(f)|^2 + |D(f)|^2} \tag{4.4}$$

By dividing the numerator and denominator by the noise power spectrum, $|D(f)|^2$, the Wiener filter can be expressed in terms of the SNR

$$W(f) = \frac{SNR(f)}{SNR(f) + 1} \tag{4.5}$$

Using the Wiener filter, an estimate of the clean power spectrum is calculated as

$$|\hat{X}(f)|^2 = |\hat{Y}(f)|^2 W(f) \tag{4.6}$$

Following the same procedure as for spectral subtraction, the enhanced power spectrum can be transformed back to the time-domain through combination with the noisy phase and an inverse DFT.

4.3.3 Log spectral amplitude estimation

Several studies [122] have compared spectral subtraction, Wiener and LSA and have been in general agreement that LSA gives best performance [54, 144]. LSA uses both *a priori* and *a posteriori* estimates of the SNR to derive a filter for noise reduction. An estimate of the clean magnitude spectrum for the ith frame, $|\hat{X}_i(f)|$, is computed as

$$|\hat{X}_i(f)| = \Big[\frac{\xi_i(k)}{1 + \xi_i(k)} \exp\Big(\frac{1}{2}\int_{v_i(k)}^{\infty} \frac{e^{-t}}{t} dt\Big)\Big] |Y_i(k)| \tag{4.7}$$

where $|Y_i(k)|$ is the noisy magnitude spectrum and $v_i(k)$ is defined as

$$v_i(f) = \frac{\xi_i(f)\gamma_i(f)}{1 + \xi_i(f)} \tag{4.8}$$

Variables $\xi_i(k)$ and $\gamma(f)_i$ are, respectively, the *a priori* and *a posteriori* estimates of the SNR, calculated as

$$\xi_i(k) = \zeta \frac{|X_{i-1}(f)|^2}{|\widehat{D_{i-1}}(f)|^2} + (1 - \zeta) \max[\gamma_i(k) - 1, 0] \tag{4.9}$$

and

$$\gamma_i(k) = \frac{|X_i(k)|^2}{|\widehat{D}_i(k)|^2} \tag{4.10}$$

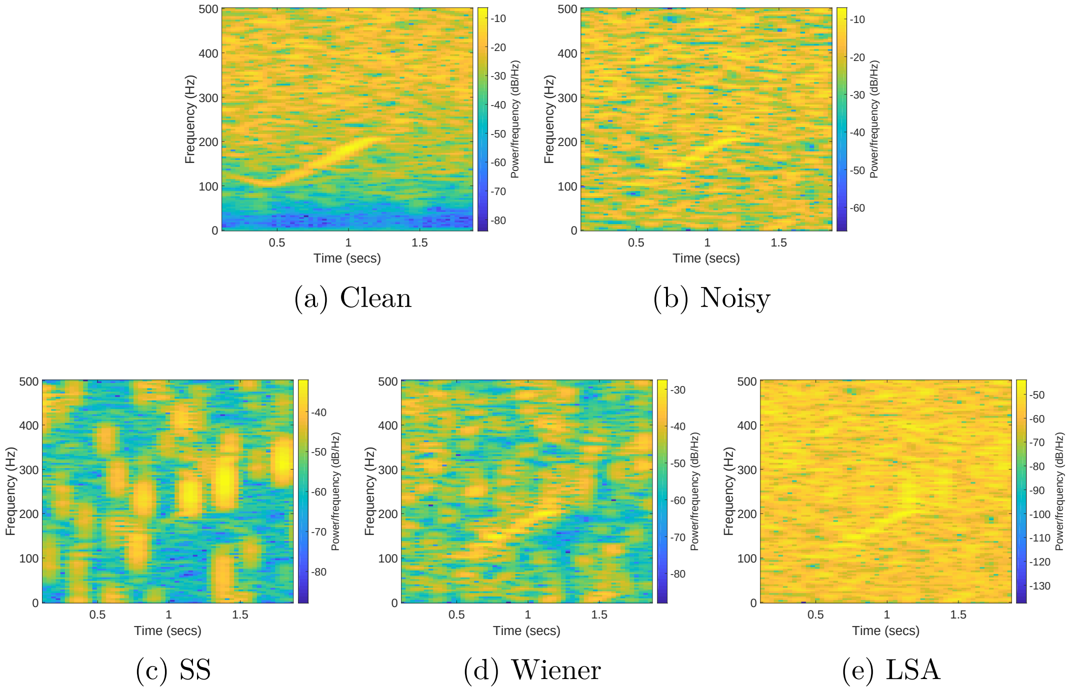
The filtering methods all require statistics of the noise and many methods have been proposed to provide these and include voice activity detection, minimum statistics and speech presence probability [130, 165, 197, 67]. In this work, an assumption is made that the first 100ms of audio contains only noise and the noise spectral estimate was taken by averaging noise-only vectors extracted from this region.

Spectrograms of all the audio enhancement methods described can be seen in Figure 4.3 with clean and noisy spectrograms shown for comparison. Visually, both SS and Wiener introduce a large number of artefacts into the spectrogram. LSA provides no noticeable visual enhancement to the NARW upcall. Wiener filtering looks to expose the NARW upcall the most albeit with artefacts, with LSA and SS struggling to enhance the signal.

4.3.4 Experimental setup

The audio enhancement methods discussed previously (in Section 4.3.1, 4.3.2 & 4.3.3) are now compared to assess which performs best to reduce noise for NARW detection. Detection accuracy from the proposed CNN system is established after each audio enhancement method is applied, to understand how well each method works at reducing noise and improving accuracy.

Tests in this section consider a noisy environment to evaluate the performance of the audio enhancement methods. The Cape Cod corpus, detailed in Chapter 2.6.3, is used both in the raw form (labelled as clean, to differentiate from the set with additive



(a) Clean (b) Noisy

(c) SS (d) Wiener (e) LSA

Fig. 4.3 Spectrograms of a NARW upcall in clean and noisy conditions (top), compared to audio enhancement methods (bottom).

noise) and with the addition of white noise across all samples. White noise was added to the data at an SNR of 0dB to simulate a noisy ocean environment, this process is shown in Chapter 2.6.4. An accuracy of 96.29% is established from the clean Cape Cod corpus when no enhancement is applied. When white noise is added at 0dB and tested against the original clean-trained model the accuracy drops to 80.62%. Performance improves when the model is re-trained using matched noisy data for training and testing, achieving an accuracy of 87.46%.

Two conditions are considered when evaluating the audio enhancement methods, clean-trained detector or vestigial-trained detector, with all tests reporting accuracy of the proposed CNN system after audio enhancement has been applied. First, test data is passed through the audio enhancement method under evaluation and detection is made using a clean-trained model. This model is trained with no artificial addition of noise and such is called, CLEAN. This test indicates the ability of the audio enhancement method in a real-world scenario where only a prior clean-trained model is available. Large decreases in accuracy under this condition indicate that the method has not been successful at removing noise without damaging the original signal.

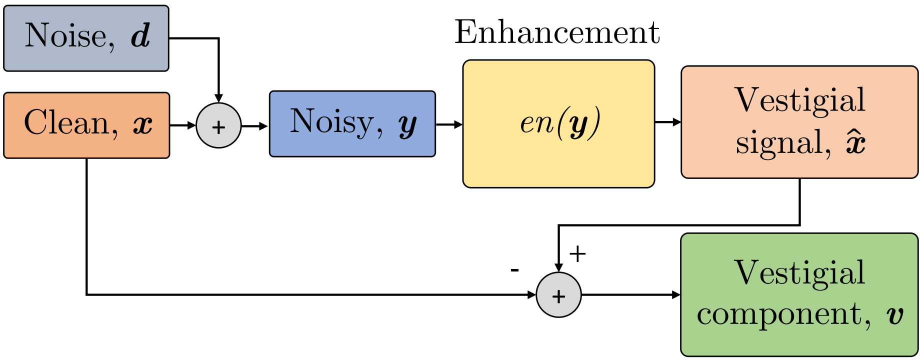


Fig. 4.4 A diagram to show the processing pipeline of the vestigial signal, $\hat{\mathbf{x}}$ and to represent the vestigial component, $\mathbf{v}$.

A second novel approach is proposed to address this mismatch. Specifically, audio enhancement is applied to the noisy training data and a new detection model is trained. The audio enhancement produces an enhanced signal, $\widehat{\mathbf{x}}$, which aims to be as similar to the original clean signal, $\mathbf{x}$ as possible. In theory, the closer the enhanced signal, $\widehat{\mathbf{x}}$ is to the clean signal, $\mathbf{x}$, the more noise has been reduced. However, after enhancement a *vestigial* component, $\mathbf{v}$, remains on the clean signal. The vestigial component, $\mathbf{v}$, represents the difference between the clean signal $\mathbf{x}$ and the enhanced signal, $\widehat{\mathbf{x}}$ where

$$\mathbf{v} = \widehat{\mathbf{x}} - \mathbf{x} \tag{4.11}$$

These tests are designed to show the maximum performance in optimal conditions where it is appropriate to retrain a model under a new condition using data containing a vestigial component. Figure 4.4 shows this processing pipeline. For clarity, the vestigial tests refer to the enhanced samples, $\hat{\mathbf{x}}$ which contains the clean signal, $\mathbf{x}$, and the vestigial component, **v**. Detection models trained with this signal are therefore referred to as VES. When new detection models are trained for VES tests, only data passed through the enhancement method under test is used to train the model.

4.3.5 Results

Experiments first examine the effect of applying audio enhancement methods on NARW detection accuracy and consider both CLEAN and VES trained CNN models for detection. To simulate noisy conditions, white noise at an SNR of 0dB is added to the audio (shown in Chapter 2.6.4) and noisy samples are processed using the audio enhancement methods. Figure 4.5 shows a comparison of the three methods within this section and baseline performance when using no enhancement method. As previous discussed, CLEAN tests use a model trained on the original, non-corrupted Cape Cod corpus. VES tests use a new model trained on data after enhancement has occurred. The first two bars on Figure 4.5 refer to tests with no enhancement (N/E). The singular MATCH test is a new model trained on the corrupted corpus without any enhancement applied.

Figure 4.5 shows that performance in noisy conditions, when using no enhancement method (N/E), drops from 96.29% to 80.62%. This drop of ~16% is not recovered by any of the audio enhancement methods with all three achieving a lower accuracy. This is attributed to the large volume of artefacts introduced by each method, which are shown in Figure 4.3. Performance in VES conditions are however improved with Wiener attaining an accuracy of 87.01%, similar to the noisy MATCH condition at 87.46%. None of the audio enhancement methods however, exceed performance when no enhancement is applied. These results establish that traditional speech and audio enhancements are not suitable enhancements for NARW vocalisations. Listening to

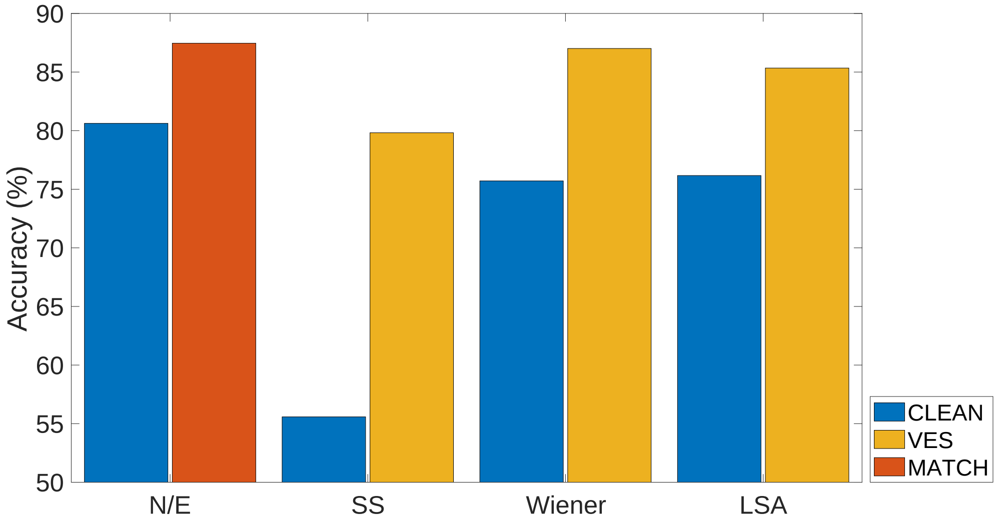
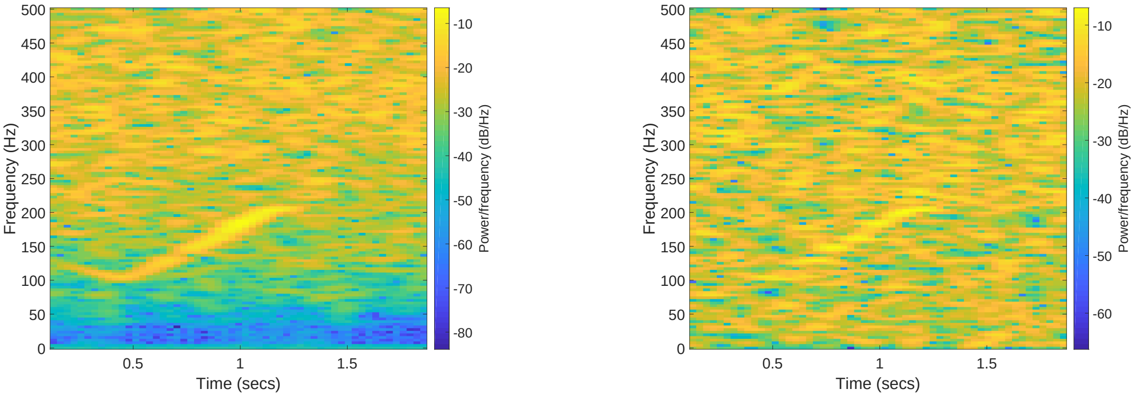


Fig. 4.5 A bar chart showing the accuracies of the NARW detection system using audio corrupted with white noise at 0dB. Accuracies show performance when no enhancement (N/E) has been applied compared to the application of various audio-based enhancement methods.

the enhanced audio also indicates that these methods are unsuccessful at reducing noise within the segments as enhanced recordings sounded more distorted than they did with the white noise at 0dB. All audio enhancements presented rely on a noise estimate to attempt enhancement, if this included part of the signal or was disproportionately high in amplitude this could have caused more severe noise removal than intended and removed parts or the entirety of the NARW vocalisation. Alternatively, as shown in Figure 4.3, all methods introduced artefacts from the enhancement process which mask the NARW vocalisation and can cause the signal to sound distorted.

4.4 Image enhancement methods

In this section, spectrogram features, first discussed in Chapter 3.3.1, are now used as input into image enhancement methods. When observing spectrograms, the presence of background noise can obscure the vocalisation of NARWs which reduces detection accuracy. Figure 4.6 demonstrates the visual characteristics of an upcall in relatively clean (Figure 4.6a) and noisy (Figure 4.6b) conditions. Figure 4.6b shows that when noise corrupts the signal the vocalisation can become masked and therefore harder to detect. This is supported by the decrease in detection accuracy from 96.29% in clean conditions to 80.62% in noisy conditions reported in Section 4.3.5



(a) Clean

(b) Noisy

Fig. 4.6 A comparison of a single NARW upcall in both clean and noisy conditions. (a) A spectrogram of an upcall with no addition of noise taken directly from the Cape Cod corpus. (b) A spectrogram of the same upcall as seen in (a) with the addition of white noise at an SNR of 0dB.

Image enhancement methods are now investigated, with the aim of enhancing the NARW upcall for more accurate detection in noisy conditions. Specifically, a range of point processing operations and histogram equalisation methods are considered. These methods were chosen as they are unsupervised and require no prior information for their application. Furthermore the chosen methods are commonly used for image enhancement tasks [115, 66, 184] and consequently used in this section to investigate the enhancement of spectrogram images. The spectrogram features used are created using the parameters outlined in Chapter 3.3.4 with tests finding these spectrogram parameters to provide the best detection accuracy for NARW vocalisations.

4.4.1 Point processing

The point processing enhancement methods investigated are the logarithm, exponential and power law [125]. These operations are applied to each point in each spectrogram feature, $X(i,j)$ to give the enhanced images, $\mathbf{X}^{log}$, $\mathbf{X}^{exp}$ and $\mathbf{X}^{\gamma}$.

$$X^{log}(i,j) = log_e(\alpha X(i,j) + 1) \tag{4.12}$$

$$X^{exp}(i,j) = e^{\beta X(i,j)} - 1 \tag{4.13}$$

$$X^{\gamma}(i,j) = X(i,j)^{\gamma} \tag{4.14}$$

Figure 4.7 shows the effect of these operations on a NARW upcall in both clean and noisy conditions. The same mix of white noise at an SNR of 0dB is used for evaluation, matching that seen in Section 4.3. The logarithm and power law ($\gamma = 0.2$) expand lower energy regions and compress higher energy regions and serve to highlight the vocalisation. The exponential and power law (γ=2.0) perform in an opposite manner which can make the upcall more difficult to observe. For testing, α and β were set to one, while γ was varied from 0.1 to 5. These values were chosen as they give a wide envelope to analyse performance of the methods, specifically both expanding and compressing values within the spectrogram.

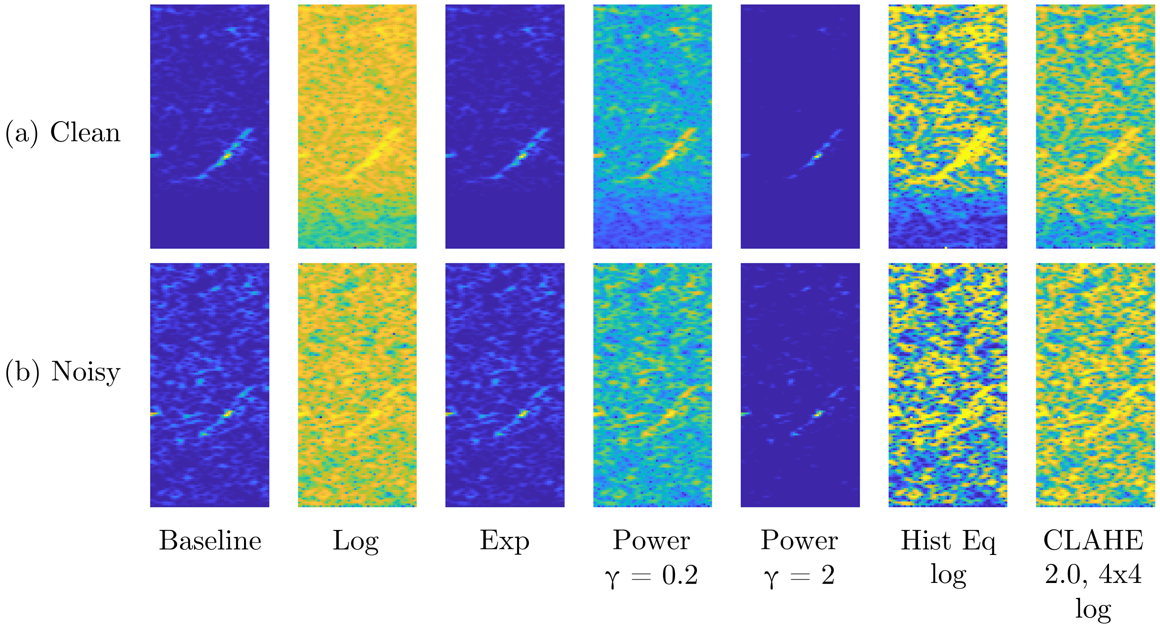


Fig. 4.7 Examples of clean and noisy spectrograms with a range of unsupervised image enhancement methods applied. Row (a) displays the clean spectrograms whilst row (b) displays the noisy. Row (b) spectrograms contain white noise added at an SNR of 0dB. Each spectrogram shows two-seconds of audio with a sampling frequency of 1kHz.

From visual inspection, Figure 4.7 shows the logarithm and power law (γ=0.2) to enhance the upcall more than the other point processing methods. All point processing methods visually struggle to enhance the upcall in noisy conditions.

4.4.2 Histogram equalisation

Two methods of histogram-based enhancement are applied. The first is standard histogram equalisation (HE) that aims to increase the global contrast of an image by flattening the distribution of pixel values [135]. This process can improve visual details in harsh conditions where parts of an image are under or over exposed. However this method increases overall contrast within the image and can intensify unwanted pixels leaving artefacts [161]. The effect of this on NARW upcalls is shown in Figure

4.7 which illustrates that more spectral detail becomes visible as the range of energy values is expanded. HE spectrograms in Figure 4.7 do however suffer from heightened exposure of background noise.

The second method is contrast-limited adaptive histogram equalisation (CLAHE) [245]. This divides the spectrogram into a grid of patches and performs localised histogram equalisation. Bilinear interpolation is then used to remove artefacts at patch borders. A contrast limiting value is used to clip histogram bins at a specific level to reduce the over exaggeration of noise within the image. CLAHE was originally developed to improve upon some of the issues that were faced when using HE, such as over exposure of pixels that were insignificant. By performing localised operations, background noise could not theoretically be exposed past the maximum of the surrounding pixels. CLAHE with a 4×4 grid is illustrated in Figure 4.7 and shows a more even distribution of energy values across the spectrogram than compared with HE, due to local time-frequency regions being processed separately. A clip limit is set when applying CLAHE and restricts values from exceeding a boundary, to limit overexposure of histogram peaks within the image. Preliminary tests found that a constant clip limit of 2.0 gave best results and so is used for all further CLAHE tests. CLAHE spectrograms in Figure 4.7 show a higher level of exposure control whilst providing less upcall enhancement than HE. Both histogram methods shown in Figure 4.7 indicate the best visual performance in noisy conditions over the other methods under test.

4.4.3 Experimental Setup

Experiments now consider a range of point processing and histogram parameters for image enhancement spectrogram representation of NARW vocalisations. The following tests in Section 4.4.4 investigate 15 parameter sets against a baseline performance when no enhancement has been applied. Tests follow a similar framework to those in Section 4.3.5 with both CLEAN and VES models being evaluated. A third set of tests are also introduced to evaluate performance in purely clean conditions, MATCH-CLEAN, with the original Cape Cod corpus not containing additive white noise for training or testing. MATCH-CLEAN tests are investigated, as image enhancement methods can be combined with the extraction of spectrogram features to provide the classifier more robust training data. Audio enhancement methods, however are specifically targeted at reducing noise and therefore are not appropriate to use in MATCH-CLEAN conditions.

Table 4.1 provides a breakdown of the training and testing data for each method. Grey cells indicate where the enhancement under test has been used. MATCH-CLEAN tests use a model trained on clean data (the same Cape Cod corpus detailed in Section 4.3.4), and test with clean data after having the enhancement under test applied. CLEAN tests use the same model as MATCH-CLEAN, but instead use noisy data at a 0dB SNR for testing. This test data first has the enhancement under test applied prior to detection. In VES conditions tests use a vestigial model trained on the noisy data at a 0dB SNR after enhancement has been applied. The same test data as used for CLEAN is also used to test VES.

Similarly to Section 4.3.4, noisy Cape Cod data was used to test the effectiveness of the image enhancements in noisy conditions. White noise was added to the data at an SNR of 0dB to simulate a noisy ocean environment, this process is described in Chapter 2.6.4 and uses the same data for tests as that described in Section 4.3.5.

Model	**Training Data**	**Testing Data**
MATCH-CLEAN	Clean	▨ Clean
CLEAN	Clean	▨ Noisy 0dB
VES	▨ Noisy 0dB	▨ Noisy 0dB

Table 4.1 Description of each test scenario. For each scenario the data used for training and testing of the model is shown. Grey cells indicate where the enhancement method under test has been applied.

4.4.4 Results

Experiments first examine the MATCH-CLEAN condition to analyse performance in clean conditions without the addition of noise. Table 4.2 shows a full breakdown of all the test parameters and results. In MATCH-CLEAN conditions, the logarithm operation achieves the highest overall detection accuracy with 96.29% of 2-second audio segments correctly classified as either NARW or not. Both HE and CLAHE (2.0 4x4) using log spectrograms as input and power ($\gamma = 0.1$) also perform well, correctly classifying over 96% of NARW upcalls, with all methods improving upon the baseline accuracy of 93.91%. In the spectrograms from Figure 4.7, the expansion of lower energy regions made by using $\gamma = 0.2$, had the effect of highlighting the NARW vocalisation. Conversely, exponential and power law methods with $\gamma > 1$, all degrade accuracy.

For tests in noisy conditions, using the CLEAN and VES models, Wiener filtering has been included in Table 4.2 as it achieved the highest accuracy in VES conditions in Section 4.3.5 and serves as a comparison to the image enhancement methods. Tests show a large drop in accuracy from MATCH-CLEAN to CLEAN indicating that all methods struggled to enhance the upcall or that they introduce artefacts into the image pushing the clean and noisy domains further apart. Table 4.2 shows HE achieves the highest detection accuracy for both CLEAN and VES tests. In both conditions multiple methods achieve near maximum performance (CLAHE 2.0 4x4, HE and log)

Method	MATCH -CLEAN	CLEAN	VES
Baseline	93.91	80.47	87.65
Logarithm	**96.29**	80.62	87.46
Exponential	92.38	80.41	86.69
Power ($\gamma = 0.1$)	96.03	80.91	87.57
Power ($\gamma = 0.2$)	95.98	81.37	87.84
Power ($\gamma = 0.3$)	95.80	80.82	87.71
Power ($\gamma = 0.5$)	95.50	81.16	87.98
Power ($\gamma = 0.7$)	95.06	80.40	87.94
Power ($\gamma = 2.0$)	89.78	80.50	85.66
Power ($\gamma = 5.0$)	80.59	78.39	78.12
HE	94.70	80.44	87.82
HE (log)	96.04	**81.59**	**88.18**
CLAHE 2.0 4x4	93.73	79.84	87.96
CLAHE 2.0 16x16	93.84	77.42	87.64
CLAHE 2.0 4x4 (log)	96.14	80.86	87.22
CLAHE 2.0 16x16 (log)	95.70	79.02	83.58
Wiener	–	75.71	87.01

Table 4.2 NARW detection accuracy using image enhancement methods in clean and noisy conditions.

and show performance can be improved greatly by re-training with the vestigial signal.

4.5 Discussion

In this chapter audio-based and image-based enhancement methods have been applied to NARW vocalisations to investigate potential improvements to detection accuracy in noisy conditions. White noise was added to the original corpus at an SNR of 0dB to simulate a noisy ocean environment and to cause broadband corruption. An SNR of 0dB provides a noise environment with equal power of signal and noise components. Simulating this noise environment more realistically tests the noise reduction methods for their ability to enhance the detector in harsh conditions.

Initially, audio-based methods, previously used for speech enhancement, were investigated for their potential to improve detection accuracy. Tests evaluated detection accuracy of the proposed CNN system against both a clean-trained and the proposed vestigial-trained model. Results found that applying spectral subtraction, Wiener filtering and log spectral amplitude methods decreased accuracy in noisy conditions compared to performance when no enhancement had been applied. This is attributed to the addition of artefacts within the audio introduced by the audio enhancements. Due to the introduction of artefacts, it is suggested that the enhanced spectrograms were less visually recognisable compared to the original noisy spectrograms, hence the reduction in accuracy.

Further experiments investigated image-based enhancements, exploring point processing and histogram equalisation methods. Image enhancements were applied to the spectrograms of each two second audio block. Tests found that applying image-based enhancement to spectrogram features was able to improve accuracy in both CLEAN and VES conditions. Of the methods tested, log and histogram equalisation on log features were found to perform best. Baseline performance, when no enhancement had been applied, achieved near highest accuracy in both CLEAN and VES conditions. Taking the logarithm of the spectral values provides a similar performance to the baseline in both CLEAN and VES tests with a 2.38% increase in accuracy in MATCH-CLEAN conditions.

Further tests investigating noise reduction of NARW vocalisations should utilise log or log-HE spectrogram features as these outperformed baseline accuracies across all tests. The methods tested within this chapter, although preliminary, highlighted the best performing feature extraction pipeline for further noise reduction tests. Future work will investigate more advanced methods of noise reduction that aim to be more suitable for denoising NARW vocalisations in harsh conditions.

Chapter 5

Application of autoencoders for denoising North Atlantic right whale vocalisations

5.1 Introduction

This chapter proposes a robust solution for detecting North Atlantic right whales in noisy conditions by developing autoencoders methods for noise reduction. Both autoencoders (AEs) and denoising autoencoders (DAEs) are investigated for reducing noise in ocean recordings before detection using the proposed CNN system detailed in Chapter 3.3.4. Autoencoders [174] are a methodology developed to encode an input feature by restricting the available latent space between input size and output size. Restriction of the latent space forces data to be lost during encoding. The theory behind autoencoders is to learn the values that represent the feature and only propagate these through the latent space. Although information is lost during encoding, it is intended that the retained data should provide a feature, once decoded, that defines the class label more closely than prior to encoding. Training an autoencoder allows for output features to reflect on their origin in order to more closely resemble the original feature, with irrelevant information, such as background noise, compressed. Denoising autoencoders [217] utilise the same data structure albeit with noisy-clean feature pairs. Noisy features are used as input, with clean features as the target that are used to optimise the DAE output. During the DAE training, the output feature is compared to the clean feature with the model parameters updated such that the output more closely resembles the clean target.

In recent years, autoencoders (AEs) have gained popularity as a data denoising technique. Originally, they were investigated for use as a compression algorithm [203] but their performance did not match that of specific compression models, such as JPEG or MP3 [6]. Autoencoders first require a model to be trained to effectively learn compression of the input. They often work best in applications with similar targets, such as images of the same object where a pattern can be learnt, instead of broad applications such as generic image compression. For this reason autoencoders are a good candidate for enhancing NARW call spectral images, as dedicated training for the model is possible. Figure 5.1 shows how the proposed autoencoder fits into the current CNN detection system (Chapter 3.3.4). Noisy inputs are fed into the AE with denoised outputs being used for detection via the CNN. When noise corrupt data is applied to an uncompensated detector, the accuracy falls substantially. This was previously shown in Chapter 4.4.4 where accuracy of the detector without noise was 96.29%, and dropped to 80.62% when noise was added. In order to build a more robust solution to detecting NARW in noisy conditions, both autoencoders and denoising autoencoders are applied to denoise noisy spectrogram representations of NARW vocalisations.

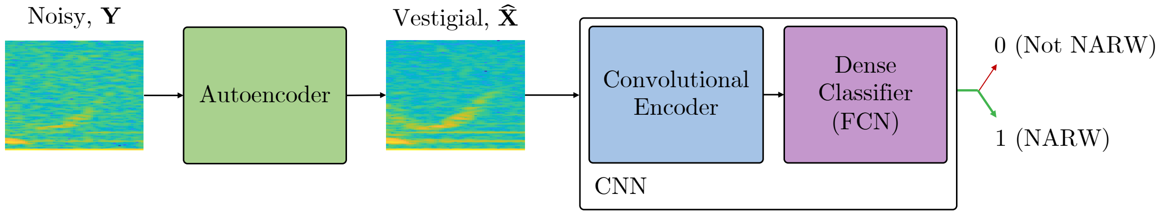


Fig. 5.1 Diagram of the denoising detection system, showing how the autoencoder attaches to a CNN.

The remainder of this chapter is organised as follows. Section 5.2 gives a background into the origin of autoencoders, how they operate and previous work. Section 5.3 introduces the proposed autoencoder and denoising autoencoder systems for noise reduction. In Section 5.3.1 standard autoencoders are tested as a pre-processing stage to improve detection accuracy in clean conditions, aiming to suppress unwanted ambient noise whilst emphasising the NARW vocalisation. Section 5.3.2 presents denoising autoencoders (DAEs) which are investigated with the aim of reducing additive anthropogenic noise. Preliminary experimental results are presented in Section 5.4, with tests investigating the most suitable model architecture for autoencoders in Section 5.4.1. Lastly, tests to explore the effectiveness of the DAE are then presented in Section 5.4.2 and evaluated using a range of model architectures.

5.2 Background

Autoencoders [174] are a method of data compression and they aim to learn the most efficient encoding [88] of an input feature, $\mathbf{x}$, by compression into a latent space, $\mathbf{b}$. Producing the most efficient encoding is done by maintaining values that represent the class. Class defining values are discovered by evaluating many input features to learn which elements appear often. For NARW detection this would be propagating the vocalisation, whilst ignoring values that change from input to input such as background noise. Once compressed the feature is decoded to produce an output, ψ, similar visually to the input. Autoencoders are a neural network architecture with layers used to compress the input feature (encoder). Figure 5.2 shows an example AE architecture using fully connected layers. Figure 5.2 uses one encoding layer between the input, $\mathbf{x}$, and the latent space, $\mathbf{b}$. The latent space (often called a bottleneck if smaller in size than the input) lies equally between the input and output. It is responsible for holding the most compressed representation of the input feature before being decoded into an output feature. As discussed previously for other neural networks in Chapter 3.3.2, AEs also use a period of training to learn the weights and biases for each layer within the network.

For detection, a NARW vocalisation appears in every positive detection event. Using an AE as a pre-processor aims to encode the class defining features into the bottleneck before decoding back into the output. During decoding, the network can only use bottleneck values and weights and biases to up-sample the bottleneck feature in size. This means that any values not found in the bottleneck cannot be reproduced into the output feature. This can suppress background noise that changes for every input and correctly learn the pattern of the NARW vocalisation for propagation. Ultimately this should produce a cleaner (less noisy) output feature that still resembles the original vocalisation, with the benefit of creating a more distinct feature for detection. As previously seen in Chapter 4.4.4, high levels of noise cause a drop in detection accuracy and therefore highlight the potential benefit of denoising methods that can successfully reduce noise to subsequently make detection easier.

Autoencoders use the same processing structure as other neural networks, albeit with some layers reversed for decoding. The architecture in Figure 5.2 shows the encoding function, $enc()$ and decoding function $dec()$ of an AE. The middle layer,

b, provides the bottleneck feature. Equation 5.1 & 5.2 show how the input feature, **x**, is processed to produce the output feature, ψ. After an output is produced, the network calculates a reconstruction loss by evaluating the original input, **x** against the reconstructed network output, ψ. Throughout the training process the reconstruction loss is used by an optimisation function to update layer weights, which in turn aims to minimise the reconstruction loss, this is consistent with the process used in Chapter 3.3.2, for training other neural networks.

$$\mathbf{b} = enc(\mathbf{x}) \tag{5.1}$$

$$\psi = dec(\mathbf{b}) \tag{5.2}$$

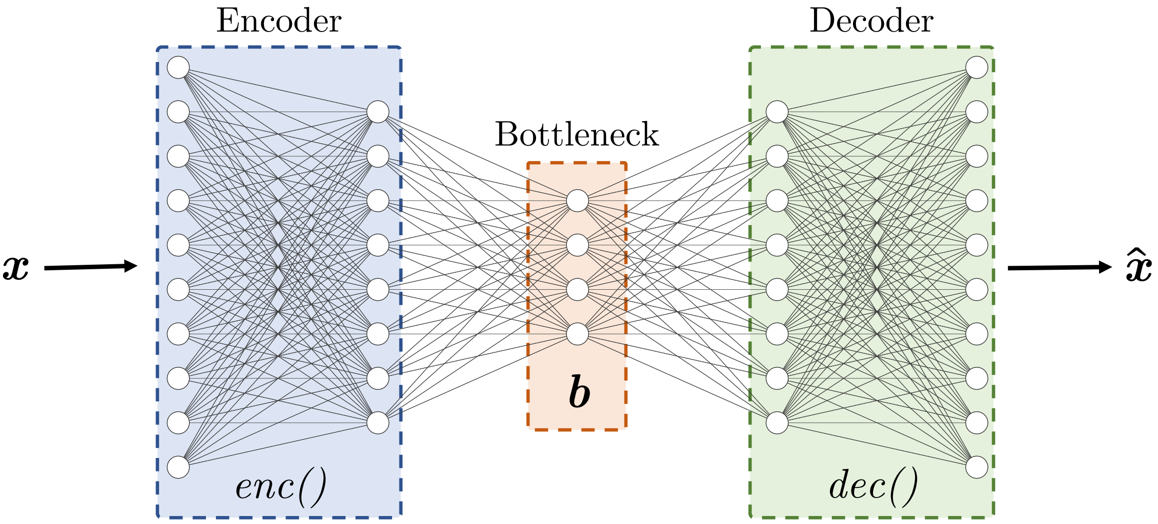


Fig. 5.2 Example fully connected autoencoder architecture with one encoding layer.

Denoising autoencoders, a development of autoencoders, first gained popularity after [217] showed their potential in 2008. Vincent et al. explored the development of autoencoders by testing their hypothesis to see if partially corrupted inputs could be recovered via noisy-to-clean network mappings. The architecture of this approach is similar to that of the standard autoencoder but signifies a shift from a compression based theory to one of denoising. Vincent et al. [217] uses the MNIST dataset for testing and applied a destruction principle to set a proportion of each image to a value of zero. Tests varied the amount of destruction from 10% to 40% and found the DAE to outperform other techniques, such as support vector machines (SVMs) and deep belief networks (DBNs) when classifying the MNIST digits. This work shows early potential for the use of DAEs in denoising applications. Vincent et al. continued their work in 2010 with further investigation into DAEs [218]. They explore the potential of stacking DAEs (SDAEs) to create deep architectures for more complex datasets. Their work however is notably different from creating deep DAEs. SDAEs are a series of stacked shallow DAEs whilst deep DAEs contain multiple hidden layers. The work presented within this chapter refers specifically to DAEs and expands on traditional AEs that use a single hidden layer, and explores the use of multiple hidden layers, instead of stacking shallow networks such as seen in [218]. Throughout their experiments in [218], SDAEs consistently perform better than SVMs and DBNs producing a lower error rate across nearly all tests.

Autoencoders have previously been applied to variety of sound and image applications, ranging from speech enhancement [123, 5] to sound event classification [243, 170] and medical imaging [234, 241]. In 2013, [123] used denoising autoencoders for speech enhancement. They used noisy-clean speech pairs, transformed into spectrograms, to train a DAE to filter speech prior to speech estimation. Estimating clean speech from noisy is an important task in speech recognition. Often, operating conditions of speech recognition systems are less than ideal and subsequently noisy environments are common. For example, the latest Google Assistant speech recognition systems need to operate under conditions with music, background noise (TV, hairdryer, kettle boiling etc) or with competing speakers. Speech recognition systems therefore need robust techniques to deal with noise. Ocean sounds are similar in principle, and classification methods work best under clean conditions with a high SNR. Similarly to human speech, ocean based acoustics can often suffer from corruption. Other animals, shipping, construction or a range of other anthropogenic noises can interfere or even completely mask the signal of interest [212]. Consequently a robust system to deal with variation in input data is essential for effective operation. Lu et al. found that using a DAE gave higher performance when compared to using the minimum mean square error (MMSE) for speech enhancement [123]. Using the DAE produced similar spectral images to the clean data, and based on the testing results, this provides compelling evidence for the use of DAEs in denoising applications.

In 2016 [73] used DAEs for denoising of medical images. Image enhancement and denoising is broad area of research with a range of applications. Medical imaging is similar to other types of imaging and is susceptible to noise [178]. A limitation when capturing images is the exposure (or lack of) to light. Bad lighting conditions or environments with low light often produces images that are underexposed. Modern cameras can artificially add electrical gain (ISO) to increase exposure when the image is captured [102]. In medical imagery the process is similar when professionals attempt to decrease the patients exposure to radiation [73]. In such cases, gain is increased to compensate for the reduction in radiation granularity and hence exposure across the frame. Gain is a uniform amplification of the imaging equipment signal which allows for a better exposed image but can introduce artefacts, often detailed as noise across the image. In ideal conditions, increased gain would not be required, however when conditions are poor, gain can provide vital exposure for the medical professional to correctly analyse the image. Reduction of noise through DAEs can therefore be invaluable to situations such as these. Noise addition to acoustic recordings can produce a similar effect when viewed as spectrograms. Depending on the additive noise, corruption across the spectrogram can be visible in the image, making detection of the signal significantly harder, especially for images with low signal levels prior to corruption. [73] found using a DAE to be effective at denoising medical images. Example images show that visually the DAE has been effective with results comparing a range of techniques, with the DAE outperforming a median filter and a non-local mean filter by a large margin. [73] also supports the use of a DAE for noise removal for noisy NARW spectral images.

AEs and DAEs have never been previously used for NARW detection nor in an attempt to improved the reliability or robustness of a detection system. Work in this chapter aims to investigate both to create a more stable platform for detection in a range of conditions, namely noisy environments where mitigation efforts may be required.

5.3 Structure of autoencoders and denoising autoencoders

This work investigates the use of autoencoders as a precursor to detection, specifically to remove unwanted background noise. Importantly, this work differs from early work in this field as deep AEs are employed instead of stacked AEs [217, 218], an example of which can be seen in Figure 5.2. Deeper AEs are used as previous work investigating neural network architectures in Chapter 3.3.4 found shallower networks to produce worse accuracy than deeper structures. Another significant difference is the use of convolutional layers instead of fully connected layers within the network architecture. Originally a single hidden layer was used however prior research shows the powerful encoding capability of convolutional layers for image recognition [106, 214]. Convolutional layers are therefore used as the intermediary network layers, as can be seen in Figure 5.3, that illustrates the architecture used.

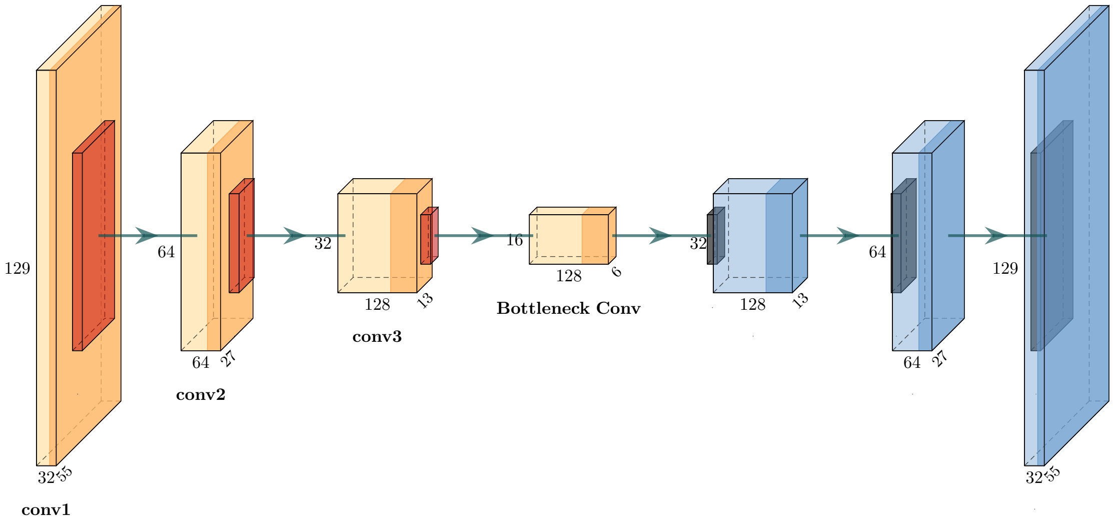


Fig. 5.3 Convolutional autoencoder architecture with three encoding layers and decoding layers. Encoding convolutional layers can be seen in light orange with max-pooling shown in dark orange. Dark blue shows convolutional layers for decoding with light blue indicating the upsampling operation.

Fundamentally, convolutional autoencoders use the same building blocks as convolutional neural networks to create models for processing input data. Autoencoders are made up primarily of an encoder, bottleneck and decoder. The encoder, responsible for compressing the input feature, directly reflects those used in the CNN. The encoding block is comprised of a series of convolutional layers. Each layer extracts information using the convolutional operation before the max-pooling operation downsamples the matrix. The bottleneck feature, found equal distance from the input and output of an autoencoder (Figure 5.3), holds the most compressed representation of the input. It functions as a standard convolutional layer in this instance however, depending on the application can be manipulated into a different shape prior to decoding. Finally, the decoding block can be described as a reversed encoding block with the first layer taking the bottleneck feature and applying a convolution operation before upsampling to a larger matrix. The upsampling layer increases dimensionality within the matrix by repeating rows and columns of the matrix in order to reach the desired width and height. Whilst encoding, network weights, **W**, on each layer, provide opportunity for the network to learn the best input feature values to propagate deeper. Similarly whilst decoding, network weights, **W**, can alter the upsampled values based on the network loss.

Tests conducted in Section 5.4.1 explore the optimal autoencoder architecture for improving detection accuracy when using a CNN. The tests vary the number of encoder and decoder layers to see the effect that network depth has on the reconstructed output. All input features are produced using the same spectrogram parameters ($f_s = 1000Hz$, $N = 256$, $S = 32$) discovered in Chapter 3.3.4, which create spectrograms of (129, 55) time-frequency elements. Tested autoencoder architectures compress these spectrograms, with the size of the bottleneck feature varying from (64, 27) elements for a one layer model, to (2, 1) elements for a 6 layer model. Smaller bottlenecks are produced when more layers are used as max-pooling operations on each convolutional layer halve the input size. It should be noted that without padding the initial input, or not using max-pooling, the architecture presented cannot use more than 6 layers as features become too small.

5.3.1 Autoencoder training

Training an autoencoder is similar to that of a CNN. However, unlike the training process of a CNN, the autoencoder does not require a separate set of distinct target data, normally termed labels and as such is an unsupervised algorithm. Instead, the data used as the target matches that of the training data. Training the model is carried out over a specified number of epochs in order to update model weights and reduce the loss.

During training, the model weights are updated via backpropagation. In order to update weight values in the correct direction, an optimiser function is used. The optimiser function, Adam [103], utilises a reconstruction loss calculated during each pass of the network. The reconstruction loss, $\mathcal{L}(\widehat{\mathbf{X}}, \mathbf{X})$, is calculated between the autoencoder output $\widehat{\mathbf{X}}$ and the target, which for this application is the clean data, $\mathbf{X}$, shown in Equation 5.3, where t and f represent the time and frequency axes respectively. $\mathcal{L}(\widehat{\mathbf{X}}, \mathbf{X})$ is generated with a binary cross-entropy loss function [43]. The aim of training the network is to minimise $\mathcal{L}$, by adjusting network parameter values, indicating that the network can effectively compress the initial input with minimal loss to quality when reconstructed. A minimal $\mathcal{L}$ value will also lead to smaller weight changes from the optimiser as the model nears minimum reconstruction error.

$$\mathcal{L}(\widehat{\mathbf{X}}, \mathbf{X}) = -\frac{1}{t \times f} \sum_{i}^{t} \sum_{j}^{f} [x_{ij} \cdot log(\hat{x}_{ij}) + (1 - x_{ij}) \cdot log(1 - \hat{x}_{ij})] \tag{5.3}$$

Once the model reaches a local minimum $\mathcal{L}$ and accuracy stabilises, training the model further is unlikely to yield significant improvements in performance. Tests described in Section 5.4 run for a fixed number of epochs to ensure the network has sufficient time to find the smallest $\mathcal{L}$ possible. All tests are given an equal number of epochs to make them comparable. Figure 5.4 shows the effect of training on the autoencoder output. The left spectrogram, Figure 5.4a, shows the noisy model input. The middle spectrogram, Figure 5.4b, shows the model output after only training for a single epoch. The right spectrogram, Figure 5.4c, shows the model output after training for 100 epochs. A vast difference between Figure 5.4b, and Figure 5.4c is noted. A lack of training time can cause the distorted affect seen, as the network is continuously

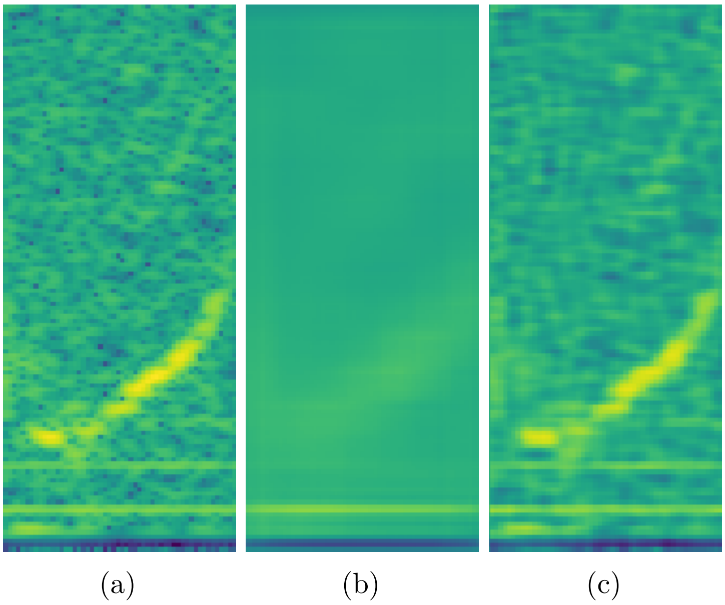


Fig. 5.4 Spectrograms of a single right whale upcall after a) prior to autoencoder, b) output from the autoencoder trained for a single epoch, c) output from the autoencoder trained for 100 epochs.

learning the correct amount to update weights to minimise the reconstruction loss. Due to the effect seen in Figure 5.4, all tests in Section 5.4 ran for 100 epochs with the best performing model saved for processing test data.

5.3.2 Denoising autoencoder training

In the previous section, the training process of an autoencoder was detailed. Denoising autoencoders (DAEs) have the same architecture as AEs, but differ in the way they are trained. Instead of utilising matched training and target data in order to generate $\mathcal{L}$, DAEs use noisy-clean pairs to train the model. Noisy-clean pairs are a produced from a single segment of audio. The *clean* segment is the original data, whilst the corrupted, *noisy* segment is the same audio as the clean with added noise. The standard AE only uses clean data to train the model. To train the DAE, the original training data is duplicated and one half corrupted with noise to create *noisy* segments. The noisy segments are used as input whilst the clean segments are used as targets or labels for the model to learn the correct output.

Tests in Section 5.3.2 initially use white noise as a corruption source due to the broadband noise that is introduced into the spectral domain. Chapter 2.6 details the process of corrupting the audio segments with noise. Further tests also explore multiple signal-to-noise ratios for corruption. This is illustrated in Figure 5.5 which shows an original spectrogram, without additive noise, compared to varying levels of noise corruption in the spectral domain. Simulating varying levels of noise corruption provides a testing framework to evaluate denoising performance across a range of SNRs. Insight from testing against multiple SNRs also provides a more accurate scenario when considering real-world performance in noisy conditions. Figure 5.5 clearly highlights the damage that high levels of noise can have when visually inspecting spectrograms of NARW upcalls. When the noise level is much higher than the signal, as can be seen in Figure 5.5e, the NARW upcall becomes disjointed and difficult to identify visually.

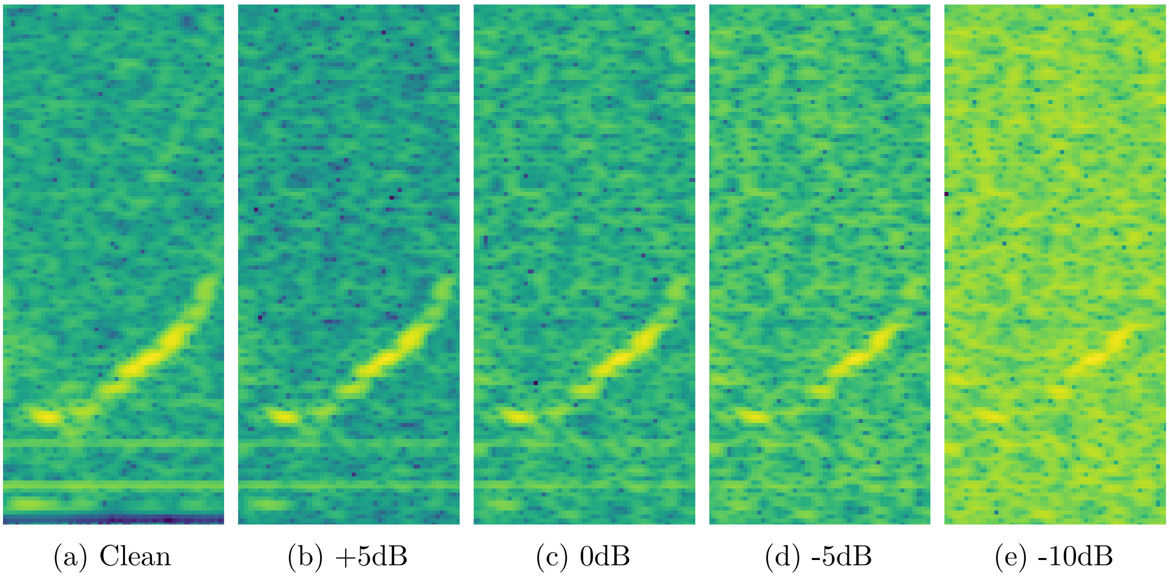

(a) Clean (b) +5dB (c) 0dB (d) -5dB (e) -10dB

Fig. 5.5 Comparison of spectrograms of a single NARW upcall when white noise has been added at varying SNRs.

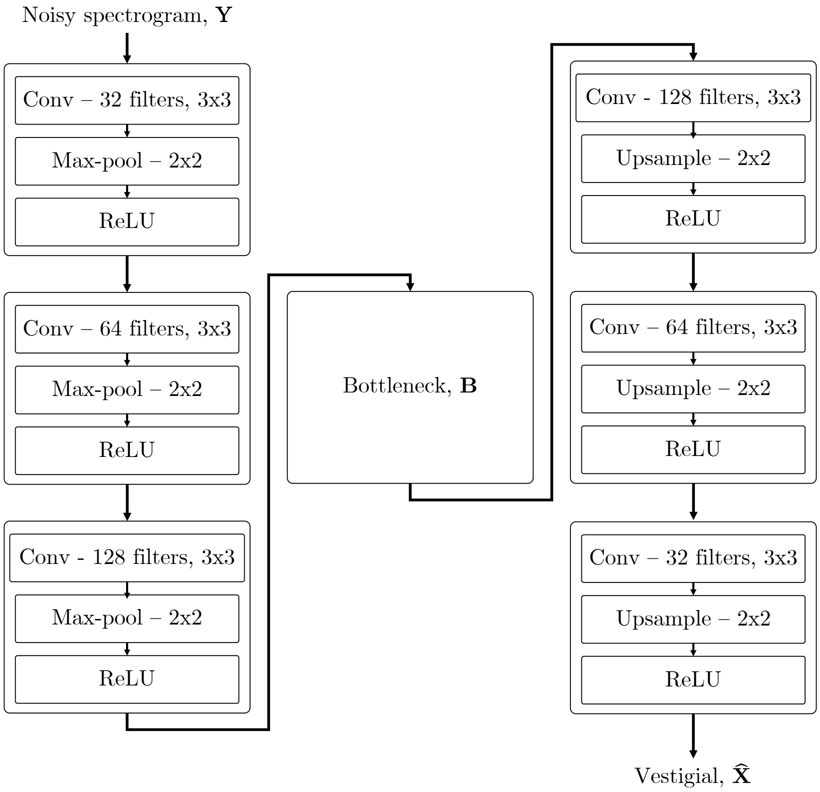
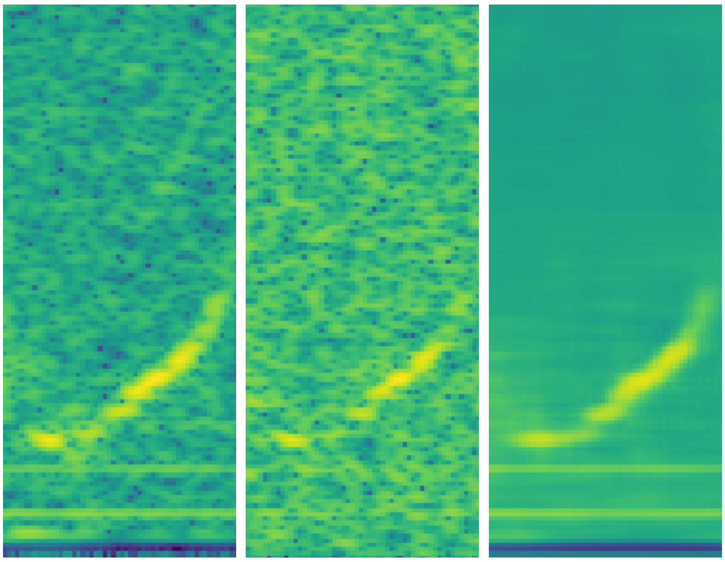


Fig. 5.6 A diagram to show the parameters of each layer in the architecture for an example three layer encoder and decoder denoising autoencoder.

The main difference between the AE and DAE is therefore the difference between the input and target data. The AE uses matched pairs, whereas the DAE uses noisy-clean pairs. The loss function shown in Equation 5.3 however is consistent with the DAE as the loss is still calculated between the clean target feature, $\mathbf{X}$, and the DAE output feature, $\widehat{\mathbf{X}}$. The difference is the DAE takes in a noisy input feature, $\mathbf{Y}$, which is a corruption of $\mathbf{X}$. An example DAE architecture can be see in Figure 5.6 with three encoder and decoder layers. Similar to the autoencoder, $\widehat{\mathbf{X}}$ represents the DAE reconstructed output. $\widehat{\mathbf{X}}$ aims to match more closely to the clean representation, $\mathbf{X}$, than the noisy representation, $\mathbf{Y}$. In theory, the closer that $\widehat{\mathbf{X}}$ and $\mathbf{X}$ are to one another, the greater the reduction of noise. In practice, the loss between $\widehat{\mathbf{X}}$ and $\mathbf{X}$ may not become as small as expected, however this does not indicate that noise has not been reduced but instead that the background noise of $\widehat{\mathbf{X}}$ does not match that of $\mathbf{X}$.

$\widehat{\mathbf{X}}$ potentially benefits from further noise reduction due to the compression effect of the autoencoder encoding and therefore should still be more suitable for classification than the original noisy spectrogram, $\mathbf{Y}$. Importantly, the NARW vocalisation is the target of the denoising autoencoder and achieving a clean reconstructed vocalisation holds more weight than $\mathcal{L}$ reaching zero. Figure 5.7 visually shows this concept, with Figure 5.7c showing the suppression of background noise over Figure 5.7b and even the clean spectrogram of Figure 5.7a. Critically however, the NARW vocalisation is preserved during the denoising process, showing the successful learning taking place whilst training the DAE.



(a) Clean (b) -5dB (c) DAE output

Fig. 5.7 Spectrograms of a single NARW upcall, (a) before corruption, (b) corrupted with white noise at -5dB, (c) output from the DAE after processing (b).

5.4 Preliminary experimental results

The aim of these experiments is three-fold. First, Section 5.4.1 explores the effect of autoencoder architecture depth and analyses the effect this has on detection accuracy of NARW vocalisations. Second, Section 5.4.2 compares performance of the DAE with the AE to assess which provides the highest detection accuracy when testing under the same noise conditions. Third, a formal analysis of the DAE is presented, exploring the DAE output during the denoising process and investigating the differences when varying DAE architecture depths. Performance of DAEs across a range of noise conditions with varying model architectures is also given to discover which is most suitable for implementation on a NARW detection system.

5.4.1 Autoencoder architecture tests

Preliminary experiments use the Cape Cod dataset (outlined in Chapter 2.6.3) to evaluate the effectiveness of the autoencoder method when detecting NARW vocalisations. These experiments were aimed to not only see if the autoencoder could be effective at suppressing background noise but also to compare a range of architectures to create an optimal model and network structure. All experiments use a convolutional encoder setup, using a convolutional layer, max-pooling layer and ReLU activation function to form a single encoding layer. This can be seen in Figure 5.3 and in Section 5.3, but the tests vary the number of encoder and consequently decoder layers. Previous work [214] established that architecture depth gave the greatest variation in network performance when comparing network hyperparameters. Varying the autoencoder depth should therefore give a strong understanding of how well the architecture can perform. Tests also use the previously proposed (Chapter 3.3.4) CNN system [212] for detection, after processing via the autoencoder, with all detection accuracies and configurations reported in Figure 5.8.

The results in Figure 5.8 show that accuracy degrades when the autoencoder architecture deepens, with all tests dropping in accuracy from 96.29% when not using the autoencoder. This result is surprising and indicates that standard AEs are not

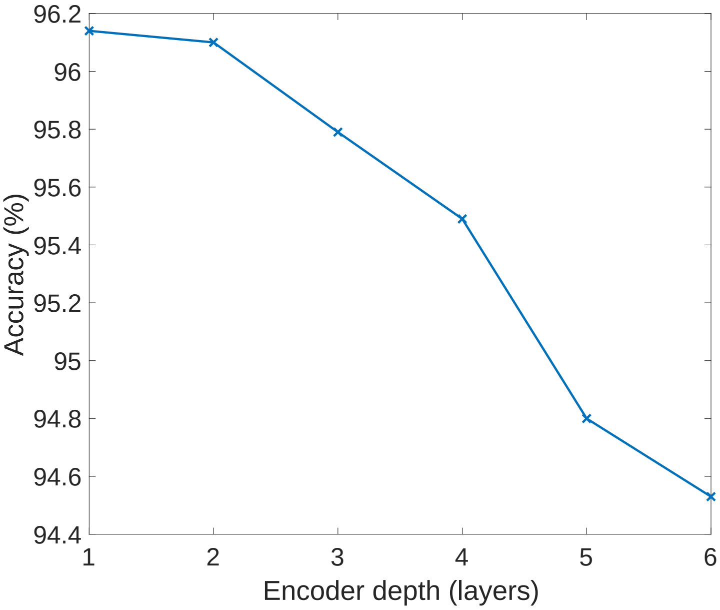


Fig. 5.8 Detection accuracies of autoencoder performance comparing a range of network architecture depths.

suitable for producing *cleaner* features for improved detection. Figure 5.9 shows two NARW upcalls before and after processing via the AE using a 3 layer encoder/decoder architecture. Both show loss of detail and contain an overall hazy quality. Figure 5.9 visually indicates why performance when using the AE has dropped slightly. Although AEs work effectively at compression of data samples, in this case they can not provide a gain in detection accuracy. It is therefore recommended that since the clean data contains little unwanted or potentially class misleading information (such as background noise), that the standard AE is not used as a pre-processing feature enhancement.

5.4.2 Denoising autoencoder architecture tests

Tests in this section now consider the denoising autoencoder across a range of noise levels for the detection of NARW vocalisations. As DAEs are aimed at creating a model to denoise the data, it is expected that losing unwanted noise comes as a consequence

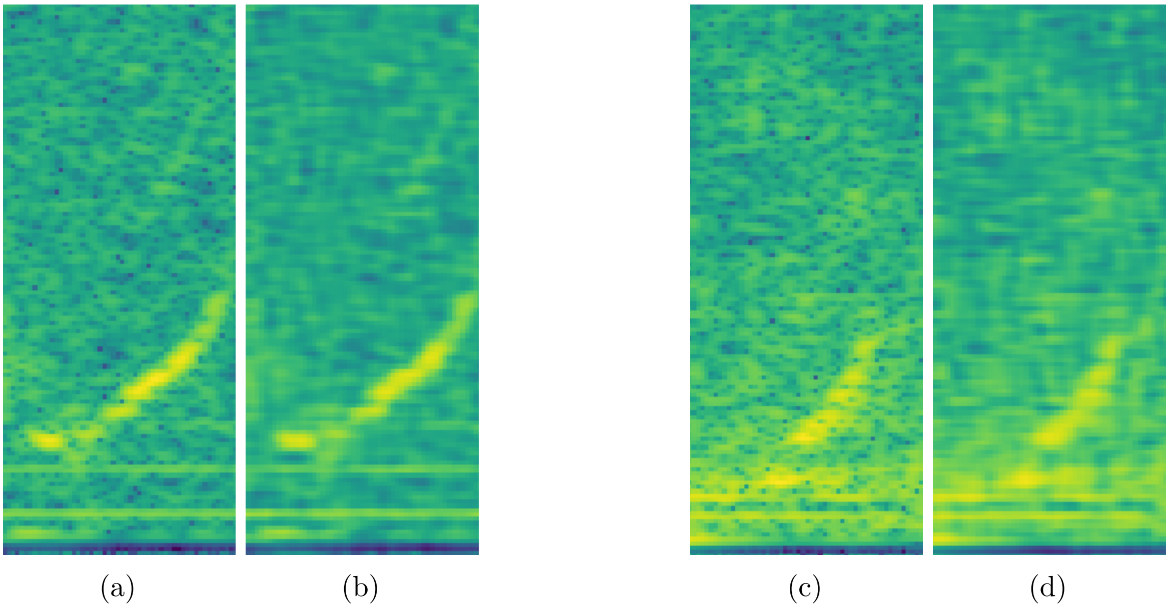


Fig. 5.9 Spectrograms of two NARW upcalls (a) & (c). Spectrograms (b) & (d) show the AE output after processing spectrograms (a) & (c) respectively. The AE to generate these spectrograms had a three layer encoder architecture.

of passing through a bottleneck and forcing a high level of data compression. This is an advantage of DAEs, however this compression may cause reconstruction artefacts or poor reconstructed representations, making further detection and classification difficult.

Tests now use the same Cape Cod corpus, however utilise a noisy alternative with additive white noise ranging from 5dB to -10dB. The processing of adding noise is described in Chapter 2.6.3. Figure 5.10 shows a comparison between detection accuracies when using no pre-processing method, the AE, and the DAE. Figure 5.10 shows the DAE achieves a higher detection accuracy over the baseline result, whilst the AE performs similarly or worse than not using the autoencoder. This test demonstrates the ability of the DAE in noisy conditions to improve detection accuracy. Further tests continue development of the DAE as it has shown to be successful at denoising corrupted signals compared to the AE. Figure 5.11 shows a grid of spectrograms prior to detection. The top row have not been pre-processed, whilst the middle row have

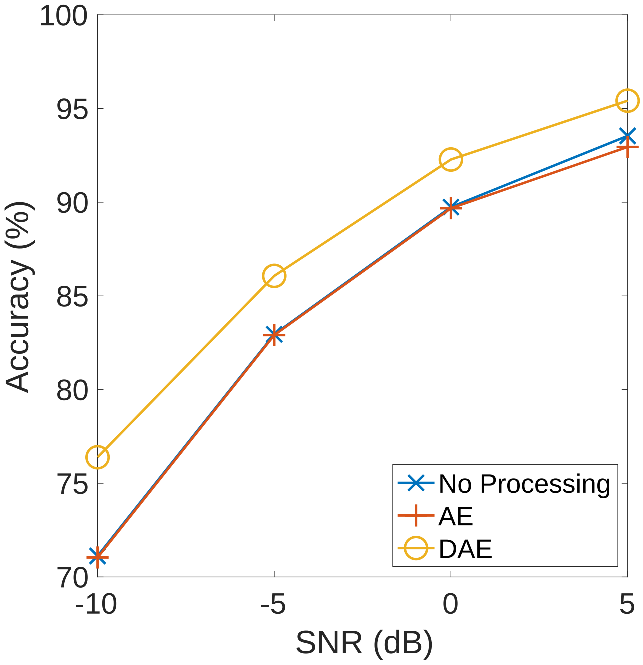


Fig. 5.10 Comparison of pre-processing techniques by evaluating detection accuracy. Original data was corrupted with white noise at 5dB, 0dB, -5dB, and -10dB. All tests used the same underlying clean model for detection and both the AE and DAE use a 3 layer encoder/decoder architecture.

been processed by the AE, and the bottom row processed by the DAE. Figure 5.11 visually highlights the performance of the DAE in all noise conditions, managing to maintain visibility of the NARW upcall when the SNR is as low as -10dB.

Tests now aim to optimise the DAE architecture. As discussed earlier, white noise at varying SNRs was added to the Cape Cod dataset before processing with the DAE. The testing framework was designed to explore a range of DAE architectures and evaluate performance across a range of noise levels. Four SNRs, 5dB, 0dB, -5dB, -10dB, were chosen as coverage for these tests. Since these experiments aim to provide a robust solution for denoising, testing across a range of SNRs gives a more accurate indication of real-world performance, and therefore an idea of which architecture would be most suitable for real-world deployment. For each SNR six tests were conducted. The DAE

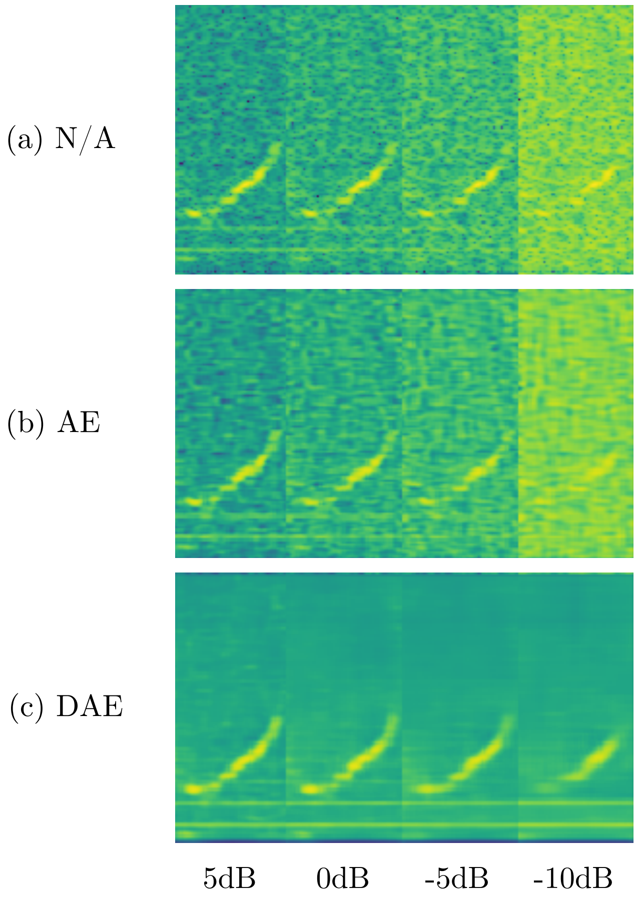


Fig. 5.11 Spectrogram evaluation of pre-processing techniques. Original data was corrupted with white noise at 5dB, 0dB, -5dB, and -10dB. Row (a) shows the original noise-corrupted spectrograms. Row (b) shows the same upcall after processing via the autoencoder. Row (c) shows the same upcall after processing via the denoising autoencoder.

architecture was varied starting at the shallowest position moving towards the deepest. In this context, shallow refers to a small number of encoding and decoding layers, with the shallowest utilising a single layer. Similarly, a deep architecture contains multiple encoding and decoding layers, the deepest possible being six. This particular test is limited to six encoding layers of depth as the initial input is (129, 55) elements in frequency and time respectively and therefore becomes too small for processing past six layers when using pooling operations.

SNR	Accuracy	Encoder depth
+5dB	90.35%	2
0dB	85.07%	3
-5dB	75.39%	3
-10dB	64.15%	4

Table 5.1 Detection accuracies across a range of SNRs with additive white noise. The architecture depth that achieved the highest result are shown for each SNR.

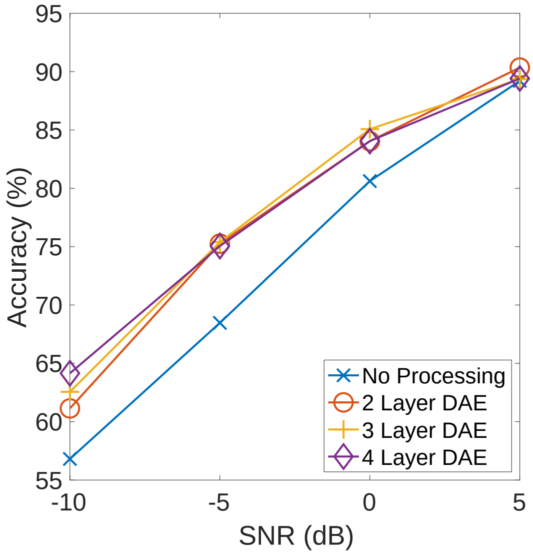


Fig. 5.12 A comparison of detection accuracy when using no pre-processing method, against using the DAE with 2, 3, or 4 layers. Tests use white noise at SNRs of 5dB, 0dB, -5dB, and -10dB.

For each SNR, table 5.1 shows the best performing DAE architecture based on detection accuracy from the CNN system. For reference, Appendix A.1 shows the extended table of results. Appendix A.1 gives accuracies for all encoder layer combinations across all SNRs. Figure 5.12 examines detection accuracy across all SNRs whilst varying the DAE architecture depth. Both Table 5.1 and Figure 5.12 indicate that the three layer DAE performed best on average across the SNR tests. Both Table 5.1 and Figure 5.12 show that the detector works best when noise is lowest, however when noise is increased a deeper DAE architecture enables a greater reduction in noise, and consequently accuracy. This is attributed to the need for deeper, more complex models to remove higher levels of noise successfully. For example the difference, Δ, at -10dB between no processing and the three layer DAE is much larger than the Δ at +5dB, indicating a higher level of noise removal.

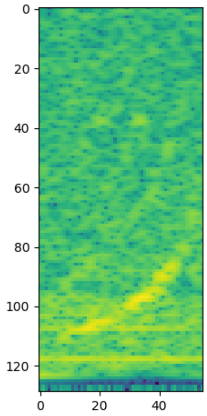


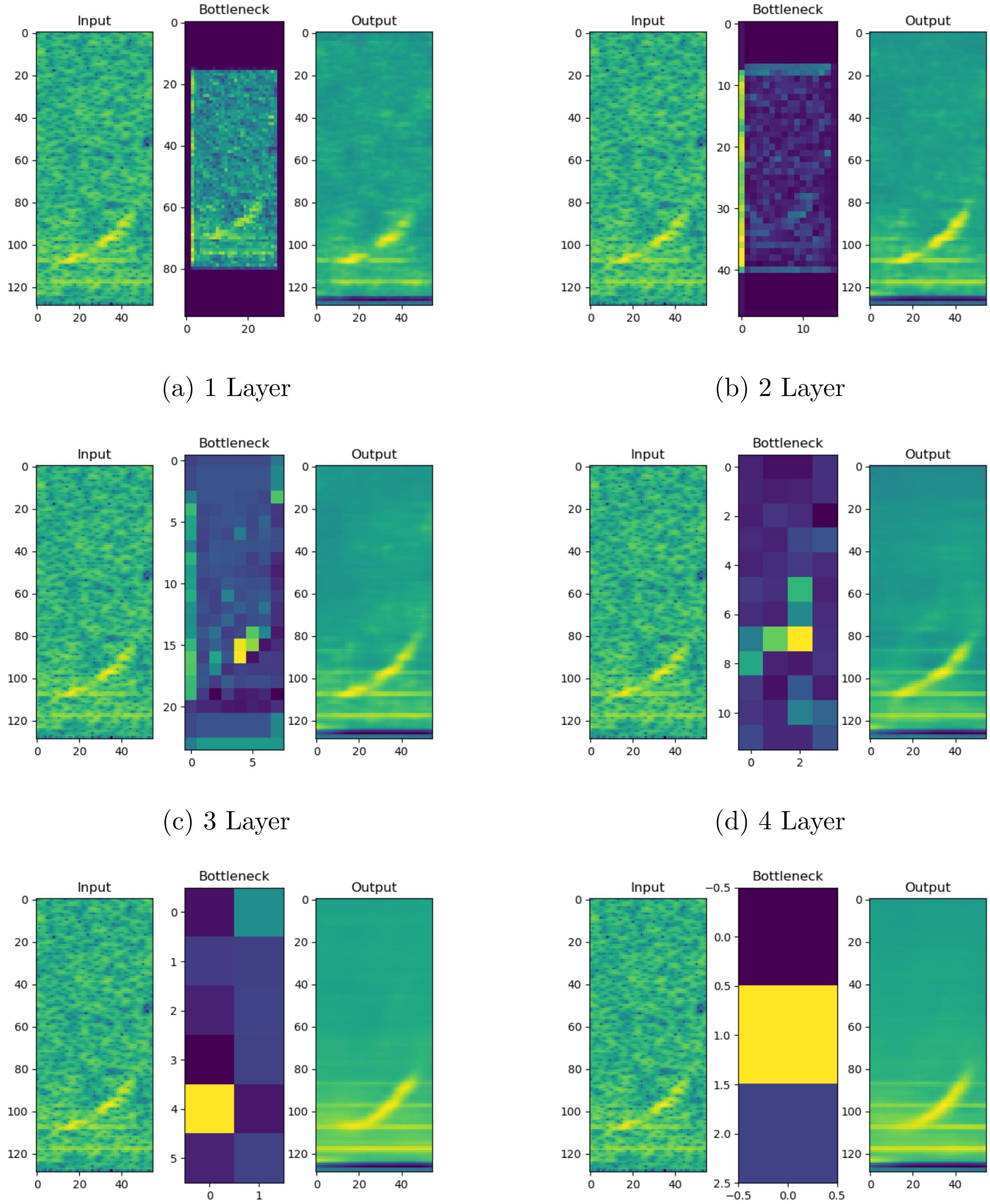
Fig. 5.13 Original spectrogram prior to noise corruption - shown in Figure 5.14.

Figure 5.14 shows examples of the DAE input, bottleneck and output features. By comparing a range of layer depths across a single SNR of 0dB, it is clear that specific architectures produce cleaner reconstructed outputs. Overall more detail is lost when using a deeper architecture as the bottleneck becomes smaller and the relative compression is higher. Even though more detail is lost, features produced by deeper models often look cleaner than shallower models - Figure 5.14a and Figure 5.14f show this comparison well. The deeper six layer architecture has more available weights to tweak during reconstruction of the image throughout training, whereas the shallower single layer architecture relies more heavily on the bottleneck feature values when reconstructing the image. Figure 5.13 shows the clean non-corrupted feature used in Figure 5.14. After analysis of results in Table 5.1 it is presumed that a mid-depth three layer architecture performs best because the output (Figure 5.14c) most closely matches that of the original clean representation shown in Figure 5.13. As Figure 5.12 shows, the DAE architecture that performs best is not consistent across the tests. However, since the three layer was the highest on average and achieved the best result in 50% of the tests, it was decided that further tests involving the DAE would utilise a three layer encoder and decoder structure.

5.5 Summary & discussion

Previous work involving autoencoders has been sparse within the domain of marine mammal detection. Other domains have benefited from research involving autoencoders such as, medical imaging [73] including histopathology image detection [234, 241], speech enhancement [123], speech recognition [5], and acoustic event classification [243]. Whilst other domains have found success with AEs, marine mammal detection has been absent from this work. Traditionally, detection of marine mammals has been manual, with many studies using basic automated detectors [211] and only recently have more modern automated methods of detection been introduced. Using autoencoders for the application of denoising marine mammal sound events is therefore in its infancy.

A range of tests have been carried out to explore the model architectures of autoencoders for pre-processing spectral images of NARW vocalisations. Tests initially used a standard autoencoder and subsequently developed into using a denoising autoencoder. Implementation of the DAE led to further exploration of noise corruption on clean signals. A range of white noise corrupted signals were created at SNRs of 5dB to -10dB. These tests aimed to provide representation in conditions more closely matched to the real-world where noise levels can vary. It was first discovered that standard AEs could not improve detection accuracy in clean or noisy conditions (Table 5.1 &



(a) 1 Layer

(b) 2 Layer

(c) 3 Layer

(d) 4 Layer

(e) 5 Layer

(f) 6 Layer

Fig. 5.14 Breakdown of the input, bottleneck and output features of a successfully denoised sample. Figures compare a number of DAE architectures, ranging from using one to six encoding layers. All figures use the same audio file with additive white noise at 0dB.

Figure 5.10), tests then focused on DAEs and understanding the relationship between detection accuracy and model architecture (Appendix A.1). Tests involving the DAE showed it to improve detection accuracy when used as a pre-processing method prior to detection. Performance on average improved by over 3% across all SNRs and consequently validated the process of pre-processing with a DAE. It is recommended that a three layer encoder and decoder DAE is used for NARW vocalisation denoising as this provided the best performance on average in noisy conditions. To further illustrate the effect of the DAE, Appendix A.2 shows spectrograms of 50 2-second blocks containing NARW vocalisations, before adding noise, after noise as been added and after the DAE has processed the noisy spectrograms.

Chapter 6

Application of convolutional neural networks for denoising North Atlantic right whale vocalisations

6.1 Introduction

This chapter aims to further explore robust noise reduction to increase detection accuracy of NARWs in noisy conditions by denoising spectrograms using a CNN to generate estimates of the noise. These noise estimates can then be subtracted from the noisy signal to provide a denoised spectrogram. This approach, named the denoising convolutional neural network (DNCNN), provides a structurally different process of noise removal compared to the autoencoders examined previously. The DAEs that were investigated in Chapter 5 are based on a neural network encoder-decoder structure aimed at compressing each input feature before expanding back to the original size without unwanted noise. In contrast, the DNCNN does not use max-pooling layers to compress the input and consequently does not reduce the input feature size when processing spectrograms. Denoising CNNs have previously been investigated for their potential improvement to classification systems in noisy conditions however the architecture and design often vary for each proposed technique [201]. Zhang et al. proposed the DNCNN architecture [239] that aims to predict the residual noise, which is said to be easier and faster to learn than a fully denoised image. The residual signal is an estimation of the noise within the original noisy signal. For denoising NARW spectrograms, this would be the noise present in a spectrogram. It is termed residual as any estimate of the noise is unlikely to produce the exact noise signal and therefore residual refers to the signal that is estimated to be the noise.

An important consideration when developing automatic detectors is the likelihood that NARW recordings will be corrupted by noise from various sources at differing signal-to-noise ratios. Depending on the distance of the NARW and location of the noise source from the receiving hydrophone, recordings can become unsuitable for detection methods, as vocalisations can be masked. Noise presents a challenge to most classification problems, from speech recognition to image identification [181, 119], and consequently many different compensation techniques have been previously proposed. These can broadly be categorised into those that attempt to match the underlying model to the characteristics of the noisy input data and those that remove noise before classification [151, 123]. This chapter solely considers the removal of noise prior to classification, with later work in Chapter 7 providing investigation and analysis of model augmentation.

The remainder of this chapter is organised as follows. Section 6.2 explains the DNCNN process and reviews literature for other denoising methods. Section 6.3 describes the process of predicting the residual noise spectrogram from the DNCNN and how this differs to predicting a denoised signal, as shown with the DAE in Chapter

5. Section 6.4 then applies the DNCNN system to NARW vocalisations, analysing a range of network parameters and their effect on classification accuracy.

6.2 Background

This section first provides a background on current image denoising techniques using the DNCNN and then details how its use of a neural network architecture is unlike other methods reviewed. Next, the process of denoising from a DNCNN is introduced with details on how the DNCNN works, with specific emphasis on how the original creators, Zhang et al. [240], intended on learning the residual signal instead of learning a denoising mapping.

Image denoising is an essential part of many computer vision processing pipelines, with removal of noise a critical task for many applications, such as for smartphone photography [2] and medical imaging [58]. The aim of image denoising is to recover or reconstruct a clean image, $\mathbf{X}$, from a corrupt image, $\mathbf{Y}$ when noise, $\mathbf{D}$ is added. Often for image denoising, during algorithmic development, $\mathbf{D}$ is assumed to be white noise. However, realistic ocean corruptions are not always broadband in frequency and are instead often transient sounds. Consequently, work within this chapter analyses acoustic recordings corrupted with white noise, with work in Chapter 7 evaluating a wider range of real-world conditions.

Currently, a range of image-based denoising methods are widely used, such as block-matching and 3D filtering (BM3D) [46], weighted nuclear norm minimisation (WNNM) [79], local spatial-spectral correlation (LSSC) [126] and non-locally centralised sparse representation (NCSR) [53]. Although the denoising potential of these methods is high, they suffer from two fundamental drawbacks. First, they are optimised to work best on Gaussian noise, commonly seen in photographic gain (a large research area of image denoising) and therefore generalisation to ocean sounds is not guaranteed. Second, each method requires optimisation and hyper-parameter searches before the optimum parameters can be used and performance can be maximised. Whilst neural networks often also require parameter setting, it is intended that once an optimum architecture is realised, further denoising will not require additional parameter changes.

The concept of the DNCNN, exploits and combines some of the most effective architectures that have been proposed for image recognition and denoising. This includes using deep architectures that are effective at increasing the learning capacity and flexibility of the model [190, 105, 240]. To improve the learning of such deep models, residual learning methods have been shown to be more effective than attempting to learn a direct mapping [82, 239]. Batch normalisation is also commonly applied and through the scaling and shifting applied at each layer, any internal covariate shift can be mitigated [72, 95]. Based on these factors, the approach taken in this chapter for denoising spectrograms of NARW vocalisations is based on a DNCNN framework that employs residual learning [239].

Convolutional neural networks have previously been discussed in Chapter 3.3.4 where they proved to give the highest accuracy for detecting NARW vocalisations. Using CNNs to extract features other than NARW vocalisations, is therefore logical and supported by the use of CNNs in many denoising applications [202, 238, 36, 112, 240]. CNNs for denoising operate similarly to those used earlier in Chapter 3.3.4 as a classifier, albeit without the need to compress the input spectrogram using max-pooling operations or to apply dense layers to produce predictions. The DNCNN stacks convolutional layers each with a ReLU activation function, to provide a network of filters which extract features from the input spectrograms. During training, the network learns the most suitable filter values for the extraction of noise features. Rather than directly outputting a denoised image $\widehat{\mathbf{X}}$, the DNCNN predicts the noise $\mathbf{D}$, the difference between the noisy spectrogram, $\mathbf{Y}$ and the clean spectrogram $\mathbf{X}$. This can be seen in Equation 6.1.

$$\mathbf{D} = \mathbf{X} - \mathbf{Y} \tag{6.1}$$

$$\widehat{\mathbf{D}} = DNCNN(\mathbf{Y}) \tag{6.2}$$

However in practice, predicting the noise can introduce artefacts of the noise removal process as well containing parts of the original clean signal. Due to this, the DNCNN predicts the noise estimate, $\widehat{\mathbf{D}}$, which is termed the residual signal. The residual signal, $\widehat{\mathbf{D}}$, aims to be as similar to the original noise, $\mathbf{D}$ as possible, as this would cause all the noise to be removed from the noisy spectrogram, $\mathbf{Y}$. Equation 6.2 details how the residual signal is produced. Unlike the DAE which predicts a denoised spectrogram, $\widehat{\mathbf{X}}$, learning the residual is faster and creates a more generic denoising model compared to predicting a more specific denoised image for each output [240]. Since a noise estimate is likely less complex than a full vocalisation spectrogram, convergence of model is likely to occur faster when attempting to predict noise. For NARW denoising, the residual spectrogram, $\widehat{\mathbf{D}}$ is then subtracted from the noisy spectrogram, $\mathbf{Y}$ to produce a clean estimate, where

$$\widehat{\mathbf{X}} = \mathbf{Y} - \widehat{\mathbf{D}} \tag{6.3}$$

The estimated clean signal, $\widehat{\mathbf{X}}$ can then be used for classification via the CNN classifier proposed in Chapter 3.3.4. Figure 6.1 shows the full denoising and classification process. In circumstances where only a clean-trained CNN classifier exists, the ability to use denoised spectrograms aims to improve classification accuracy when compared to using noisy spectrograms without denoising applied.

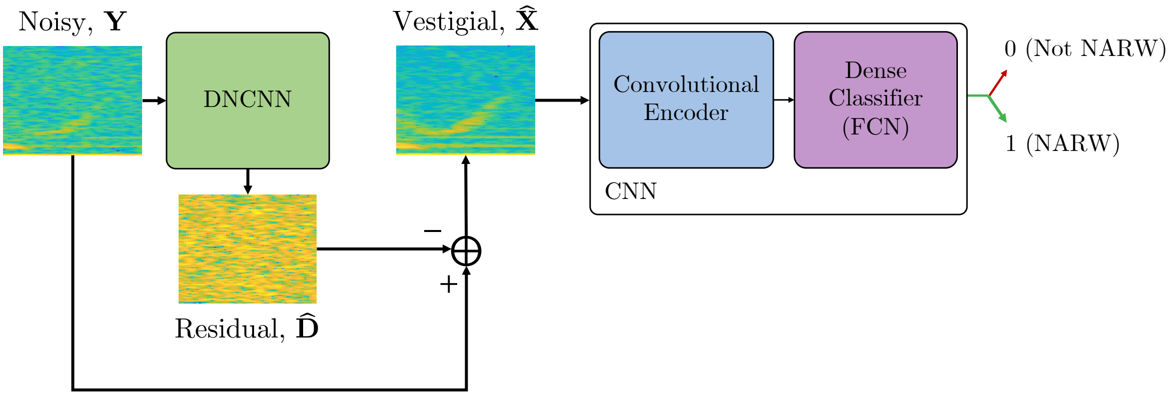


Fig. 6.1 Overview of the processing pipeline when using a DNCNN for denoising spectrograms and a CNN for classification.

6.3 Subtraction of residual noise for denoising

The aim of this section is to investigate the process of using both linear and log spectrogram features for the residual noise estimates produced by the DNCNN. Section 6.3.1 first details how noise subtraction operates when spectrograms are created linearly. Section 6.3.2 then details the process of subtracting noise estimates using log spectrograms. Finally, Section 6.3.3 provides configurations for testing both linear and log methods of noise subtraction. Analysis of the tests conducted for both are shown in Section 6.4.

In Chapter 4 a range of point processing methods were investigated to increase accuracy of the NARW detector in clean conditions. Tests found that log spectrogram features provided the highest detection accuracy in clean conditions and these have therefore been used in Chapter 5 for testing the DAE. When considering the subtraction of noise, as is required to denoise using the DNCNN, the method of noise subtraction can dramatically effect the resulting denoised spectrogram. For example linear spectrograms, can be subtracted from one another to produce the difference between them. To produce an equivalent subtraction for log spectrograms, the log operation must be accounted for. Due to this difference, both linear and log spectrograms will be considered as input features into the DNCNN.

6.3.1 Linear noise subtraction

Considering first spectrogram features that are extracted from noisy audio as described in Chapter 3.3.1 without the log operation being applied to their amplitudes. The noisy spectrogram, $\mathbf{Y}$, can be assumed equal to the sum of the clean and noise spectrograms (ignoring cross-spectral terms), $\mathbf{X}$ and $\mathbf{D}$, as

$$\mathbf{Y} = \mathbf{X} + \mathbf{D} \tag{6.4}$$

Figure 6.2 shows the full DNCNN and classification process for a linear input spectrogram. For a traditional denoising algorithm, $\mathcal{F}()$, a direct mapping from the noisy spectrogram to an estimate of the clean spectrogram would be found, i.e. $\widehat{\mathbf{X}} = \mathcal{F}(\mathbf{Y})$. Instead, when this is reformulated into the residual learning framework, a residual mapping, $\widehat{\mathbf{D}} = \mathcal{R}_{LIN}(\mathbf{Y})$, is instead learnt, where $\mathcal{R}_{LIN}()$ is the DNCNN taking a linear spectrogram. This makes an estimation of the noise spectrogram, $\widehat{\mathbf{D}}$ (i.e. residual) and when subtracted from the noisy spectrogram gives an estimate of the clean spectrogram, $\widehat{\mathbf{X}}$, as

$$\widehat{\mathbf{X}} = \mathbf{Y} - \mathcal{R}_{LIN}(\mathbf{Y}) \tag{6.5}$$

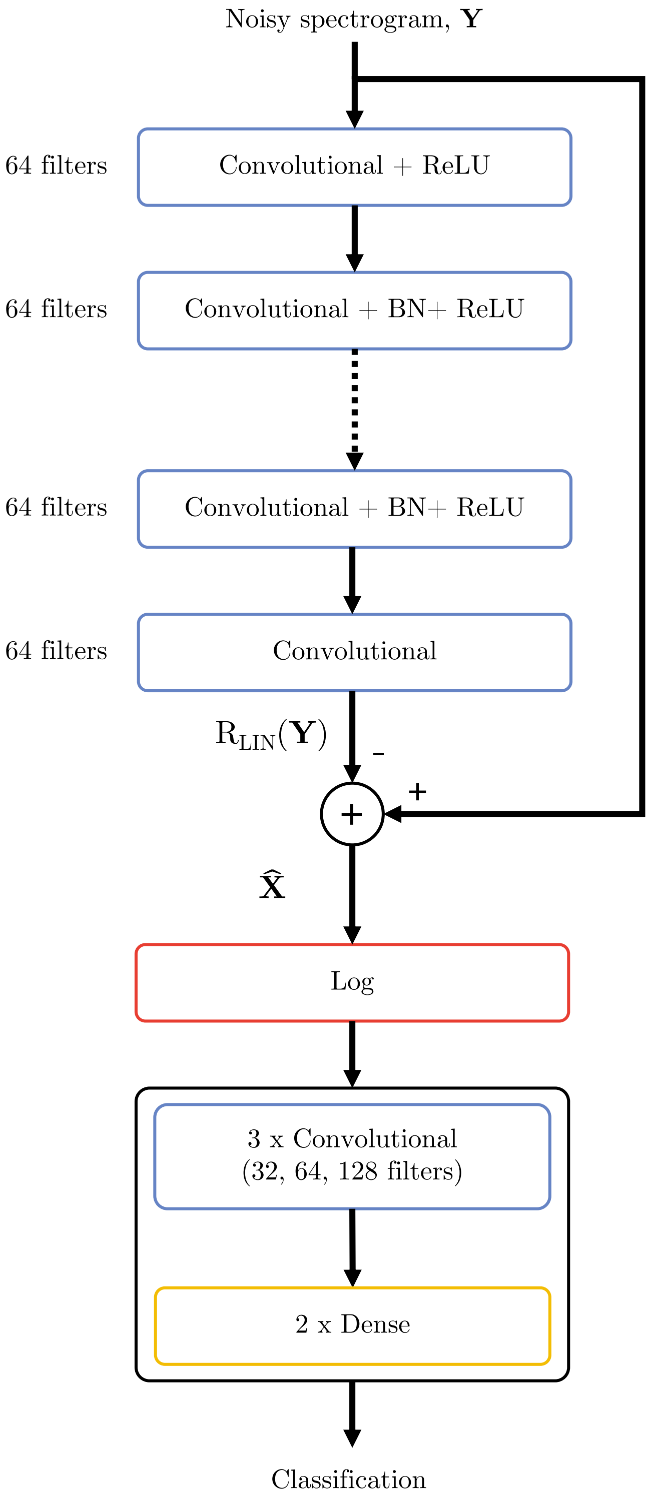


Fig. 6.2 A diagram of the DNCNN and classification pipeline using linear features for input into the DNCNN. All convolutional layers use filter sizes of 3 × 3.

6.3.2 Log noise subtraction

The alternative spectrogram feature is represented by log spectral amplitudes, which is common practice for audio processing applications. Figure 6.3 shows the DNCNN and classification process for a log input spectrogram. In this case the noisy log spectrogram, $\log(\mathbf{Y})$, is expressed as,

$$\log(\mathbf{Y}) = \log(\mathbf{X} + \mathbf{D}) \tag{6.6}$$

To calculate the same residual signal as for the linear method, a different equation is required to account for the log operation. In this case the residual signal is a subtraction of the clean spectrogram, $\log(\mathbf{X})$, against the noisy spectrogram, $\log(\mathbf{Y})$. Since the spectrograms have had the log operation applied, simply subtracting their values would result in a skewed clean estimate. To account for log, the clean estimate representation is obtained by expanding the log operation in Equation 6.6 to

$$\begin{aligned}\log(\mathbf{Y}) &= \log\left(\mathbf{X}\left(1 + \frac{\mathbf{D}}{\mathbf{X}}\right)\right)\\ &= \log(\mathbf{X}) + \log\left(1 + \frac{\mathbf{D}}{\mathbf{X}}\right)\end{aligned} \tag{6.7}$$

and so the residual mapping, $\mathcal{R}_{LOG}(\mathbf{Y})$, is

$$\mathcal{R}_{LOG}(\mathbf{Y}) = \log(\mathbf{Y}) - \log(\mathbf{X}) = \log\left(1 + \frac{\mathbf{D}}{\mathbf{X}}\right) \tag{6.8}$$

This residual signal is significantly different to that using linear spectral amplitudes in Equation 6.5 and no longer comprises just a noise component. Instead, it is a combination of the noise and clean spectrogram components.

With these two formulations for the residual, two slightly different architectures for denoising the spectrogram features are required and shown in Figures 6.2 & 6.3. Both

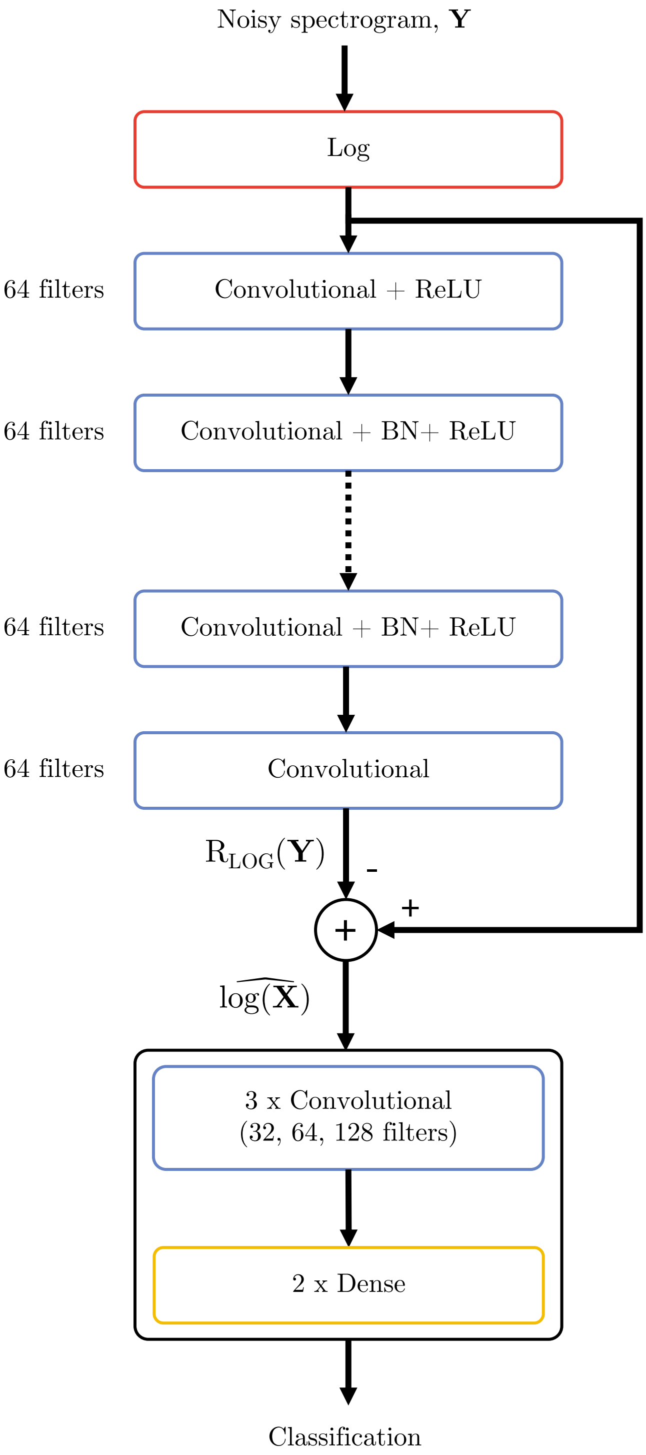


Fig. 6.3 A diagram of the DNCNN and classification pipeline using log features for input into the DNCNN. All convolutional layers use filter sizes of 3×3.

ultimately provide estimates of the clean log spectrogram for the CNN to classify.

6.3.3 Noise subtraction configurations

To perform the residual mapping, the DNCNN architecture is initially based on an approach developed for image denoising and uses a model with 17 convolutional layers [239]. The first layer has 64 filters and outputs these into a ReLU activation function [150]. The next 15 convolutional layers also use 64 filters but now incorporate batch normalisation before outputting into a ReLU activation function [72, 95]. The final layer excludes the batch normalisation and ReLU operations and outputs a prediction of the residual spectrogram elements. No pooling layers are used, so deeper models have a wider receptive field as the input is not reduced in size. With 17 layers, this corresponds to a receptive field of 35×35. For the spectrogram features this equates to a receptive field of 1.27 seconds and bandwidth of 137Hz which is broadly the duration of a NARW upcall and the frequency range of an upcall. All spectrograms are initially extracted using, $N = 256$, $S = 32$ and $f_s = 1000Hz$ which were the best performing parameters from Chapter 3.3.4.

The DNCNN is trained using pairs of spectrogram features with a noisy spectrogram used as the input and a noise spectrogram forming the training target. This matches the process used to train the DAE in Chapter 5.3.2. Noisy spectrograms for training are produced by adding the desired noise type at the required SNR to the clean time-domain signal and extracting spectrogram features. The method of noise addition was initially described in Chapter 2.6.4 and used by methods in Chapter 4 & 5. Matching the DAE in Chapter 5.3.2, white noise is used as the noise corruption in this chapter. The mean squared error is used as the loss function between the noise spectrogram and predicted spectrogram features, along with an Adam optimiser [103]. Training was performed over 50 epochs as model convergence was often fast and larger models with greater than five convolutional layers were computationally intensive.

The CNN classifier, $\mathcal{C}()$, from Chapter 3.3.4 requires a log spectrogram as input. For the DNCNN that uses log spectrogram features, $\mathcal{R}_{LOG}()$ from Section 6.3.2, the residual output is subtracted from the log noisy spectrogram to give the clean log spectrogram estimate, $\widehat{\log(\mathbf{X})}$, that is input into the classifier, where ψ represents the final classifier prediction value,

$$\widehat{\log(\mathbf{X})} = \log(\mathbf{Y}) - \mathcal{R}_{LOG}(\log(\mathbf{Y})) \tag{6.9}$$

$$\psi = \mathcal{C}(\widehat{\log(\mathbf{X})}) \tag{6.10}$$

For the DNCNN using linear spectrogram features, $\mathcal{R}_{LIN}()$, the residual output is subtracted from the linear noisy spectrogram to give the estimate of the clean linear spectrogram and this is then logged before being input into the classifier, where ψ represents the final classifier prediction value,

$$\widehat{\mathbf{X}} = \mathbf{Y} - \mathcal{R}_{LIN}(\mathbf{Y}) \tag{6.11}$$

$$\psi = \mathcal{C}(\log(\widehat{\mathbf{X}})) \tag{6.12}$$

6.4 Preliminary experimental results

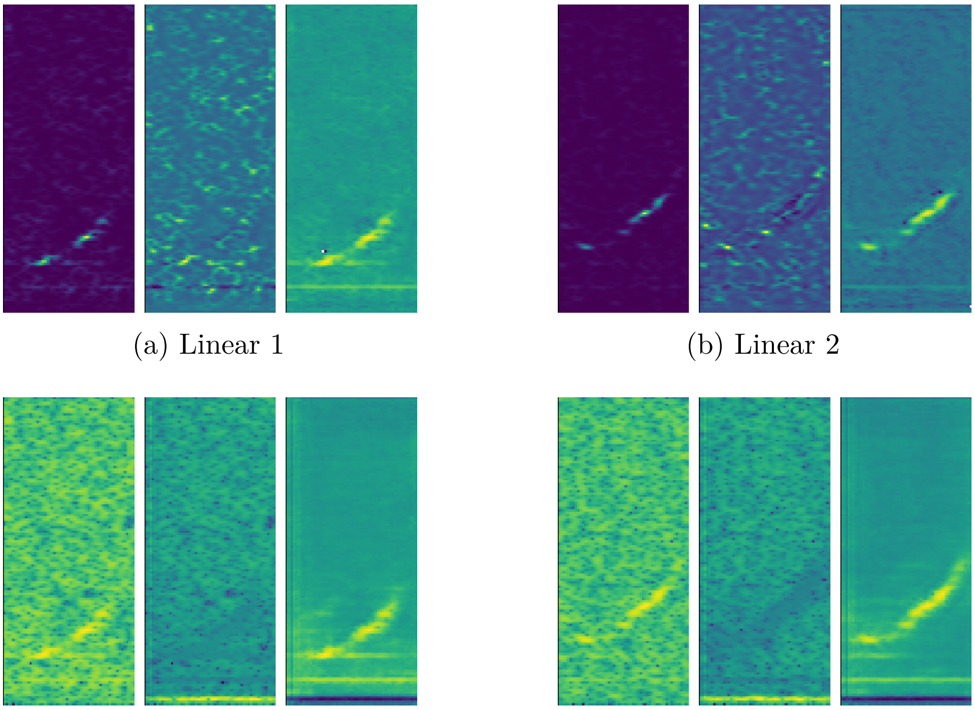
The aim of these experiments is two-fold. First, Section 6.4.1 explores the use of both log and linear spectrogram features when subtracting the residual noise from the noisy spectrograms. Second, Section 6.4.2 investigates a range of DNCNN architectures and analyses the effect this has on the detection accuracy of NARW vocalisations. Tests in Section 6.4.2 use a vestigial model, the same type used in Chapter 5.4.2. The vestigial model is trained using data that has first been denoised by the method under test. Vestigial-trained models enable the test data to match more closely to the model, meaning results can be analysed more accurately removing data mismatching as a variable of change.

All tests with this section use the Cape Cod corpus, outlined in Chapter 2.6.3, to evaluate the effectiveness of the DNCNN. Consistent with tests described in Chapter 5.4, tests used white noise at SNRs from 5dB to -10dB to corrupt the Cape Cod corpus. White noise was chosen as it provides corruption across the entire frequency spectrum and is consequently a challenging environment to denoise. If denoising improves classification accuracy in white noise conditions it suggests that the DNCNN could be successful for denoising alternative corruptions.

6.4.1 Noise subtraction analysis

As a preliminary test to establish the most suitable method of noise subtraction for the DNCNN, the accuracy of the system using log spectrogram denoising, $\mathcal{R}_{LOG}()$ was compared, with that using linear spectrogram denoising, $\mathcal{R}_{LIN}()$, shown in Figure 6.3 & 6.2. Tests used the 17 layer CNN and all training parameters were as stated in Section 6.3.3.

As an example of the visual difference between linear and log noise subtraction, Figure 6.4 shows two NARW vocalisations using both the linear (Figure 6.4a & 6.4b) and log (Figure 6.4c & 6.4d) methods. Figure 6.4a & 6.4c use the same input, similarly



(a) Linear 1

(b) Linear 2

(c) Log 1

(d) Log 2

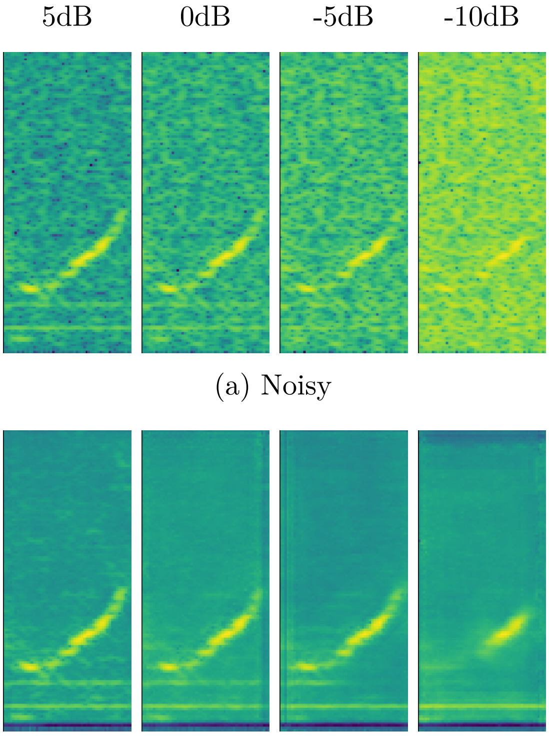
Fig. 6.4 Four figures containing the DNCNN input spectrogram (left), DNCNN output of the residual noise spectrogram (middle), spectrogram after residual subtraction from the input (right). Figures (a) & (b) show two different vocalisations using the linear subtraction method. Figures (c) & (d) show matching vocalisations from (a) & (b) respectively, using the log subtraction method. The original spectrograms are corrupted with white noise at an SNR of -5dB.

for Figure 6.4b & 6.4d. Figure 6.4 visually indicates that the log pipeline produces slightly cleaner denoised representations after residual noise removal.

Analysis of the results found an average of a 3% increase when using log spectrogram features over linear spectrogram features for all test SNRs. Tests established that using log spectrogram features for denoising achieved higher classification accuracy, which is attributed to the better conditioned spectral values the log provides, making learning the residual function more effective. For clarity, all further tests use log spectrogram features as input into the DNCNN.

6.4.2 DNCNN architecture tests

This section develops on the previously proposed 17 layer DNCNN architecture [239], exploring alternative denoising architectures and uses NARW detection accuracy to monitor performance. Further tests in Chapter 7 evaluate the DNCNN for testing in a wider range of noise conditions.



(a) Noisy

(b) Denoised

Fig. 6.5 A comparison of spectrograms in (a) noisy conditions and (b) after denoising using the DNCNN. The noisy spectrograms are corrupted with white noise at the SNRs shown.

Investigation of the original architecture is now conducted by exploring the effect of a different number of convolutional layers and convolutional filters per layer. Tests use 50 epochs to train the DNCNN and mean squared error to calculate the loss as these worked well previously in Section 6.4.1. Tests evaluate a range of network depths to find the most suitable architecture for denoising NARW vocalisations. If networks that were shallower or had less filters could achieved comparable performance to the originally proposed 17 layer architecture, it would be beneficial to use those instead as they would require less computation and subsequently less power to process each PAM recording.

Table 6.1 shows accuracy after denoising via the DNCNN using a vestigial-trained model. Although tests were run against a range of SNRs, white noise at -10dB was used for all tests in Table 6.1. An SNR of -10dB was chosen to analyse DNCNN architectures as it represents the harshest conditions for noise removal and spreads performance of each model. When conditions are less noisy many architectures performed similarly and therefore distinguishing the most suitable denoising architecture was difficult. Figure 6.5 shows an example of the denoising result using the DNCNN at all noise levels originally tested. Figure 6.5 clearly shows the more demanding conditions when the original vocalisation is corrupted with white noise at an SNR of -10dB.

As a benchmark result, an accuracy of 54.77% was attained when using a clean-trained model on the denoised data using the previously proposed DNCNN architecture of [239]. This result gives performance when the SNR is at -10dB, the most severe condition under test and the underlying model is trained on clean data. It was observed that all tests using the DNCNN on a vestigial-trained model dramatically improved performance over the clean-trained model with large increase in performance found when using both more convolutional layers and filters. This improvement is attributed to the superior mapping capability of the larger network for estimating noise. Table 6.1 identifies the best performing model to match that of [239] with 17 layers and 64 filters. Architectures with a small number of layers and filters struggled significantly.

		Filters per layer							
		2	4	8	16	32	64	128	256
Network depth (layers)	3	64.33	64.98	65.74	66.32	68.83	67.24	68.33	70.33
	5	67.17	69.85	69.80	70.00	70.17	70.00	69.83	70.33
	10	67.50	69.86	70.65	70.00	71.33	72.00	70.17	70.17
	15	68.50	70.30	70.46	70.17	72.03	72.17	70.33	70.67
	17	69.50	70.78	70.05	70.33	72.5	**72.83**	72.67	71.00

Table 6.1 Test accuracies are presented for evaluating a range of convolutional layers and filters per layer for the denoising convolutional neural network. Input data used for denoising was contaminated with white noise at -10dB.

Table 6.1 shows that detection accuracy reaches near maximum with 32 filters and peaks at 64, dropping off past 128. Architectures with 17 layers in general outperform shallower networks however accuracy using 10 layers and 64 filters is close to maximum. As the highest performing DNCNN architecture used 17 convolutional layers and 64 filters, this architecture is now used for further testing involving the DNCNN.

Figure 6.6 provides a visual comparison of four architectures tested in Table 6.1. Figure 6.6 visually demonstrates the effect of using a deeper DNCNN architecture with each increase in depth removing more noise. The 17 layer DNCNN that produced Figure 6.6e has visually the least noise and clearest NARW upcall and demonstrates why the 17 layer model provided the highest detection accuracy in tests.

6.5 Summary & discussion

This chapter has investigated the application of a denoising convolutional neural network to detect NARWs in conditions of high noise corruption. White noise was added to PAM recordings from the Cape Cod corpus at SNRs ranging from +5dB to -10dB. Two methods of DNCNN configuration where investigated. First, the method of noise

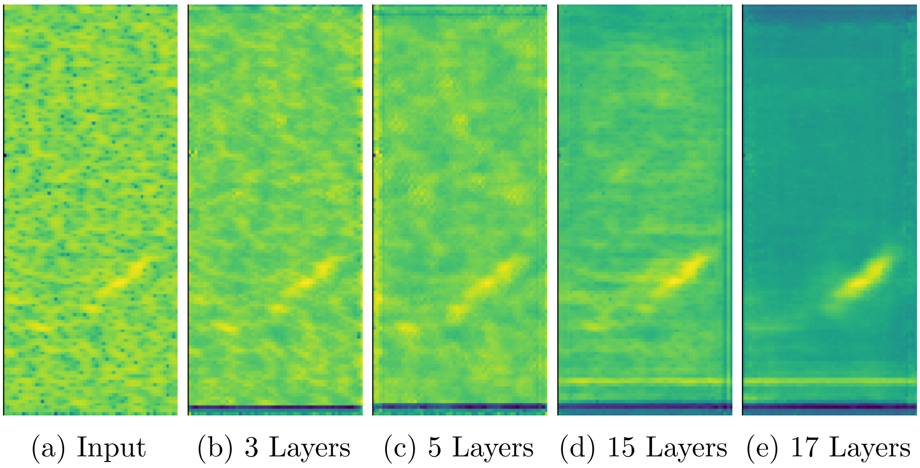

(a) Input (b) 3 Layers (c) 5 Layers (d) 15 Layers (e) 17 Layers

Fig. 6.6 A comparison of a single NARW vocalisation after processing via the DNCNN with four different network architectures. (a) shows the original spectrogram input corrupted with white noise at an SNR of -10dB. (b) used a DNCNN with 3 layers and 2 filters per layer. (c) used a DNCNN with 5 layers and 4 filters per layer. (d) used a DNCNN with 15 layers and 32 filters per layer. (e) used the best performing architecture from Table 6.1 with 17 layers and 64 filters per layer.

subtraction from a noisy input. Second, development of the DNCNN architecture to find the highest performing model. Best performance was found using log features for subtraction of predicted residual noise. Using log spectrograms instead of linear spectrograms for use in the DNCNN improved detection accuracy by a maximum of 3%. Log spectrograms were then used in further experiments to identify the best DNCNN architecture. Overall the highest detection accuracy was found using a 17 layer DNCNN with 64 filters per layer. This aligned with previous work that proposed the DNCNN by Zhang et al. [239]. Changes in the architecture all made performance worse however architectures that used above 10 layers and 32 or 64 filters still performed well. It is recommended that a denoising system for use onboard an ASV, use the 17 layer DNCNN if computational overhead allows, however a shallower 10 layer DNCNN might be more useful in more computationally restrictive situations.

Future work investigating the DNCNN for denoising NARW vocalisations could compare detection accuracy when predicting the residual noise spectrogram against prediction of a denoised spectrogram. Up until this point the residual noise spectrogram has been used to remove noise from the noisy spectrogram when using the DNCNN. Testing the inverse could provide a potentially higher detection accuracy as the DNCNN might be better at learning the underlying NARW vocalisation signal structure than the less structured noise.

Chapter 7

Investigation of neural network denoising techniques in real world conditions

7.1 Introduction

This chapter aims to further develop the denoising autoencoder (DAE) presented in Chapter 5 and the denoising convolutional neural network (DNCNN) presented in Chapter 6. This is done first in a range of noise conditions and across varying signal-to-noise ratios using the Stellwagen dataset, described in Chapter 2.6.3 with addition of noise described in Chapter 2.6.4. This investigation incorporates the CNN classifier, first proposed in Chapter 3, to develop a noise robust North Atlantic right whale classification system that achieves the highest possible classification accuracy across all noise conditions [216]. Tests introduce augmented training to explore how retraining with a new noise environment can effect detection accuracy. Tests also examine the computational speed of all proposed systems to ensure compatibility with operation from an autonomous surface vehicle as described in Chapter 2.4, is possible.

Secondly, the more naturally noisy Cape Cod corpus is used to evaluate the models developed in Section 7.3 in a blind test on a real-world noisy condition not seen by the classifier.

The remainder of this chapter is as follows. Section 7.2 describes the experimental setup of the subsequent tests, with Section 7.2.1 detailing the data and noise corruptions used in this chapter, Section 7.2.2 describes how the vestigial signal is used and the setup of both the DAE and DNCNN methods using the vestigial signal, and Section 7.2.3 provides the system configurations used to produce the experimental results. Section 7.3 presents the results of the tests explained in the previous section, across all noise types and SNRs. Specifically, Section 7.3.1 investigates augmented training across the available noise types and SNRs, with Section 7.3.2 analysing detection accuracy of both the DAE and DNCNN in the new noise conditions, Section 7.3.3 then investigates the use of augmented training, combined with the denoising methods. Finally, tests in Section 7.4 evaluate performance of the denoising methods in a new unseen condition that has naturally occurred, unlike the previously simulated noisy environments.

7.2 Vestigial classifier and denoising configurations

Work exploring the suitability of both the DAE and DNCNN methods is now carried out. In this chapter a larger range of noise corruptions are investigated to better understand how the DAE and DNCNN perform in conditions more similar to those found in the real-world. This section is used to describe the setup and structure of the tests carried out in Section 7.3.

7.2.1 Data & noise

The NARW recordings used within this section for evaluation were taken from the DCLDE 2013 workshop detailed in Chapter 2.6.3 and is referred to as *Stellwagen*. Two different problems are described in Chapter 2.6.3, a two class detection between *{not-NARW, NARW}* where either an upcall is detected or not, and a three class classification between *{upcall, gunshot, not-NARW}* where two classes represent different NARW vocalisations and third represents all other sounds. All tests prior to Section 7.4 use the three class variant and are a classification task. Tests within Section 7.4 use the two class variant as the unseen test data is from the Cape Cod corpus which only contains two classes.

The Stellwagen recordings are relatively noise-free, as example spectrograms in Figure 7.1 show, but they do contain some low amplitude noise. For the purposes of the evaluation in this chapter, the recordings are considered as *clean* and subsequently noise is added to simulate noisy audio. Given the low frequency of NARW vocalisations the audio was downsampled to 1 kHz, as previous work showed this introduces no loss in accuracy [214].

Four noise types are considered for this evaluation - tanker noise, trawler noise, shot noise and white noise. Chapter 2.6.4 describes each noise type in detail and explains how the noisy audio segments under test in Section 7.3 were created. A set of each noise type were also created for each SNR under test. In total from the original Stellwagen corpus, 16 new noisy mixtures were created for testing. Four sets for each of the four noise types at SNRs of 5dB, 0dB, -5dB, and -10dB.

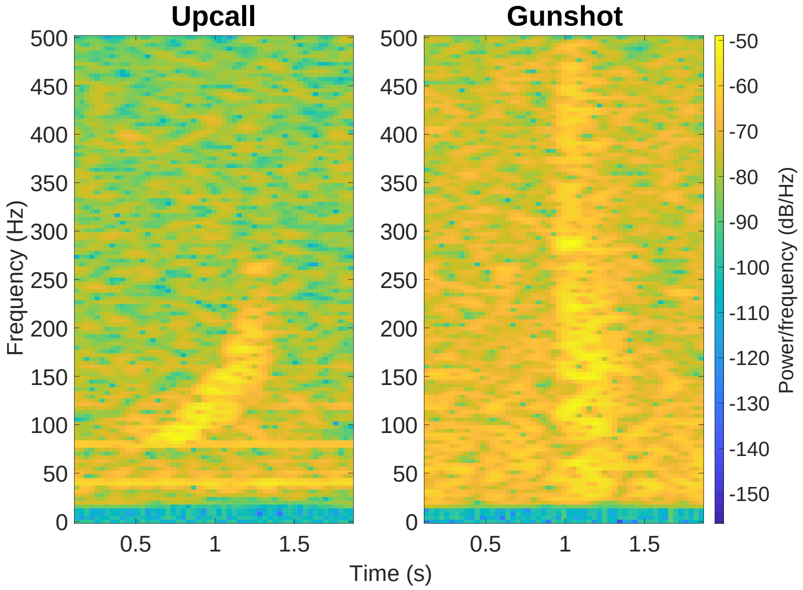


Fig. 7.1 Two example spectrograms showing a NARW upcall (left) and a NARW gunshot (right), both taken from the Stellwagen corpus.

7.2.2 Vestigial denoising setup

The vestigial, first introduced in Chapter 4.3.4 describes the signal left after denoising has taken place. As previously discussed, this is termed vestigial as the signal likely contains parts of additive noise not fully removed, and any artefacts introduced during the denoising process. For the DAE this is the spectrogram output of the DAE method. For the DNCNN this is the spectrogram produced when the residual noise is subtracted from the noisy spectrogram.

This work proposes two methods of classification using this vestigial signal. The first uses the original CNN classifier from Chapter 3.3.4, $\mathcal{C}()$ trained on clean, non-noisy spectrograms from the Stellwagen corpus. The second retrains the CNN classifier on the vestigial signal left after denoising noisy training data. This therefore matches the denoised, vestigial test data more closely. For the DAE this would create a DAE vestigial classifier, $\mathcal{C}_{DAE}()$ and for the DNCNN a vestigial classifier, $\mathcal{C}_{DNCNN}()$. Both vestigial classifiers are then trained with training data from the Stellwagen corpus first processed by the respective denoising technique.

7.2.3 System configurations for testing

The aim of the experiments presented in Section 7.3 is four-fold. First, examine the effectiveness of augmenting clean training data with noisy data for testing in noisy conditions. Second, compare classification accuracy when using training data augmentation against the explicit denoising methods of the DAE and DNCNN. Third, establish when using the DAE or DNCNN whether the classifier is best trained on clean data, $\mathcal{C}()$, or retrained on vestigial data, $\mathcal{C}_{DAE}()$ or $\mathcal{C}_{DNCNN}()$. Finally, consider how classification accuracy is affected when the noise condition in testing is unseen in training and to investigate if denoising or augmentation can improve performance in unseen conditions.

Within this section, Table 7.1 defines a set of system configurations for testing the denoising-classification pipeline. Three grouped scenarios are considered during evaluation, i) augmentation of the training data with new noise conditions with no explicit denoising applied (top four lines of Table 7.1), ii) applying the DAE prior to classification (middle four lines of Table 7.1), iii) applying the DNCNN prior to classification (bottom four lines of Table 7.1).

For each method in Table 7.1, the columns show the denoising method (i.e. none, DAE or DNCNN), the training data used for denoising (if applied) and the training data used to train the CNN classifier. The final column shows the mean classification accuracy, measured across all noise types and SNRs, and summarises the results in

Name	Denoising method	Denoising training data	Classifier training data	Mean accuracy
CLEAN	None	N/A	Clean data	72.98%
MATCH	None	N/A	Specific noise type and SNR under test	83.26%
GENERIC	None	N/A	All noise types at all SNRs	82.24%
UNSEEN	None	N/A	All noise types at all SNRs except the noise under test	72.81%
DAE-MATCH-CLEAN	DAE	Noise type and SNR under test	Clean data	82.80%
DAE-MATCH-VES	DAE	Noise type and SNR under test	Vestigial noisy data	85.18%
DAE-GENERIC-VES	DAE	All noise types at all SNRs	Vestigial noisy data	83.52%
DAE-UNSEEN-VES	DAE	All noise types at all SNRs except the noise type under test	Vestigial noisy data	73.45%
DNCNN-MATCH-CLEAN	DNCNN	Noise type and SNR under test	Clean data	79.57%
DNCNN-MATCH-VES	DNCNN	Noise type and SNR under test	Vestigial noisy data	84.71%
DNCNN-GENERIC-VES	DNCNN	All noise types at all SNRs	Vestigial noisy data	81.45%
DNCNN-UNSEEN-VES	DNCNN	All noise types at all SNRs except the noise type under test	Vestigial noisy data	72.85%

Table 7.1 Definitions of the system configurations under test. Each test specifies the denoising method and the training data used for denoising, and the CNN classifier training data. The first four methods use no explicit denoising, while the remaining methods use various configurations of either the denoising autoencoder (DAE) or the denoising CNN (DNCNN). The final column shows the mean classification accuracy of each method, taken across all noise types and SNRs from Section 7.3.

Configuration	CLEAN	MATCH	GENERIC	UNSEEN
Size of CNN classifier training set	10,000	10,000	160,000	120,000

Table 7.2 Number of training data samples used to train the CNN classifier for each method shown.

Section 7.3.

From Table 7.1, the first four configurations use no explicit denoising and instead differ in how the classifier is trained with regard to the test condition. Method CLEAN is the baseline classifier and trained on only clean training data. The classifiers used in configuration MATCH are trained on data that matches the specific noise type and SNR that is subsequently used in testing. This requires a set of 16 matched models that are used individually in each specific noise condition. Augmented training is now introduced for both GENERIC and UNSEEN methods. Augmented training expands the original Stellwagen training data to include other noise types and SNRs. The CNN classifier training set therefore becomes larger based on the number of noise types and SNRs used. The GENERIC classifier is trained on data contaminated with all four noises types at all four SNRs. This gives the most generic model for classification. For this configuration the CNN classifier will use a training set 16 times larger than the clean-trained CNN classifier. Table 7.2 shows the size of each training set. It should be noted, all augmented data is a copy of the original Stellwagen corpus and therefore no new audio segments are included. The UNSEEN classifier is similar to GENERIC, however the specific noise type under test is excluded from the training data so that the test noise condition is unseen during classifier training.

The next four methods in Table 7.1 all use the DAE for denoising prior to classification. The naming convention for these methods follows the structure DAE-<*denoising* *training data*>-<*classifier training data*>. For example method DAE-MATCH-CLEAN uses the DAE autoencoder that is trained on data matched to the specific noise test condition, with the CNN classifier trained on clean data. The denoising in method DAE-MATCH-VES is identical but the CNN classifier is now trained on the vestigial data. Method DAE-GENERIC-VES uses a DAE trained across all four noise types and four SNRs and uses a vestigial-trained CNN classifier. Finally, method DAE-UNSEEN-VES is similar except the DAE is trained on all noise types with the exception of the specific noise under test, i.e. on three noise types across all four SNRs.

The four final denoising methods in Table 7.1 use the DNCNN and have naming conventions as DNCNN-<*denoising training data*>-<*classifier training data*>. These four methods follow the same structure as those shown for the DAE.

7.3 Experimental evaluation in simulated noise conditions

The aim of these experiments is to explore the effectiveness of the DAE and DNCNN methods when denoising under different noise types and SNRs.

7.3.1 Augmented training performance

This first set of tests does not use any DAE denoising and instead examines the accuracy of the CNN classifier first proposed in Chapter 3.3.4, using the first set of system configurations from Table 7.1. The evaluation is performed across all four noise types and SNRs with classification accuracies shown in Figure 7.2. Each noise condition is evaluated using four different classification models - trained on clean data (CLEAN), trained on data matched to the specific test condition (MATCH), trained

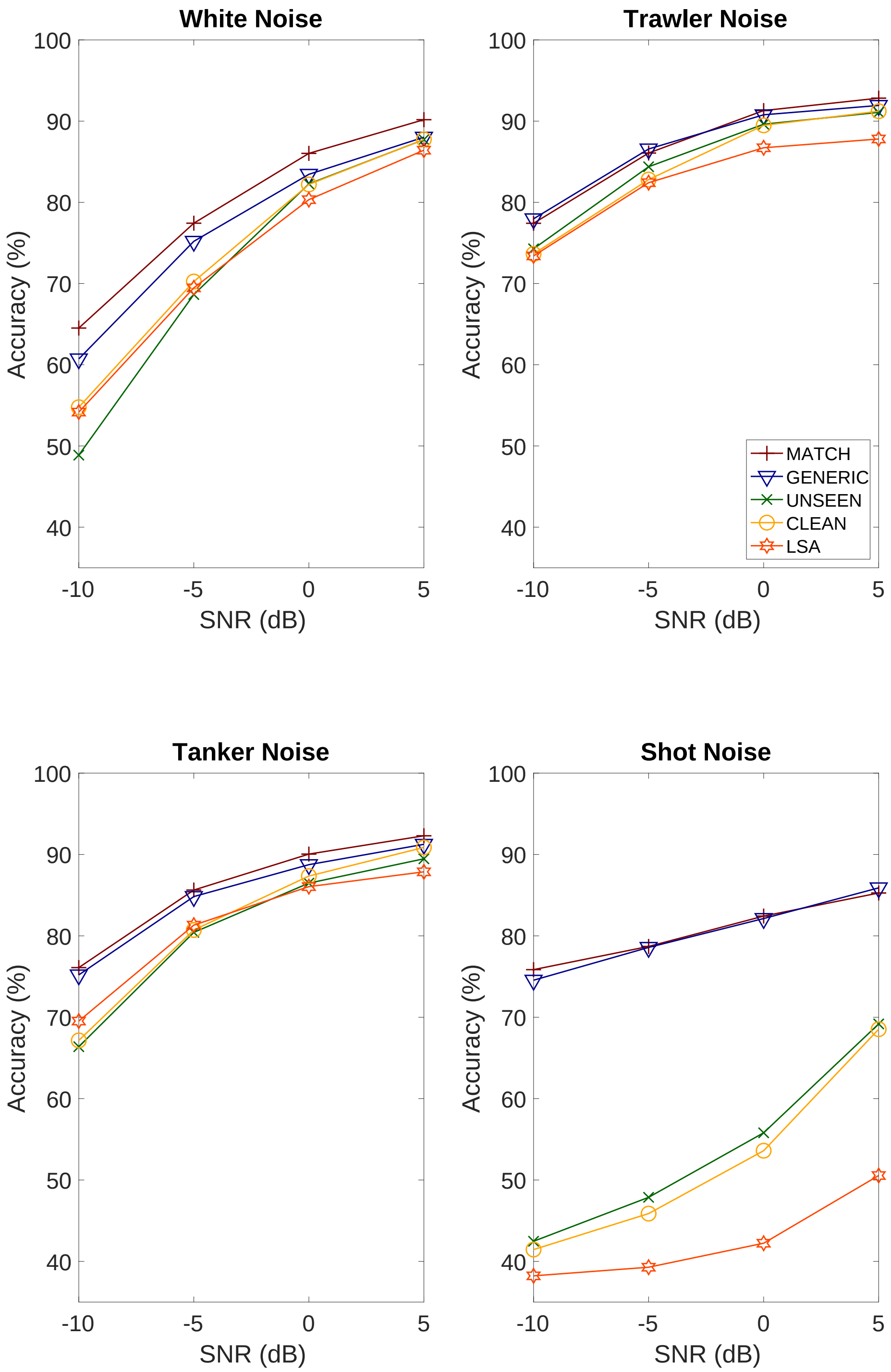


Fig. 7.2 NARW classification accuracies of the four different noise types at SNRs from -10dB to 5dB. The models are trained using different augmentation strategies with no explicit denoising, with the exception of the LSA method.

with data augmentation on all four noise types and SNRs (GENERIC) and trained with data augmentation at all SNRs on three noise conditions excluding the noise type under test (UNSEEN). To benchmark the effectiveness of these methods against an existing method of noise reduction, the log spectral amplitude (LSA) estimator was also evaluated, given its success in denoising audio signals [41]. Using the implementation in [122], the noisy examples were denoised and the resulting time-domain samples then input into the spectrogram extraction described in Chapter 3.3.4 and processed using the same CNN classifier (Chapter 3.3.4) as all other tests in this section. Classification accuracies are shown for LSA in Figure 7.2.

In noise-free test conditions the CLEAN system attains an accuracy of 94.1% but falls as SNRs reduce and in general has lowest performance. Testing using the matched model (MATCH) removes the mismatch between training and test conditions and improves accuracy substantially. However, this does require the model to be trained under the same noise conditions as seen in testing. Augmenting the training data to contain all noise types and SNRs (GENERIC) gives accuracy close to MATCH and occasionally attains higher performance which is attributed to the broad coverage of the training data. UNSEEN tests that train the CNN classifier on all but the noise type under test, reduce accuracy considerably and is comparable to the CLEAN model. LSA denoising performance is similar to that obtained using the CLEAN model, although in shot noise the performance is substantially worse. Examining spectrograms of the LSA denoised signals shows the noise to have been suppressed to a certain extent, but to now also contain short duration artefacts. These potentially cause confusion with NARW vocalisations in the classifier, particularly with upcalls, hence the inability of LSA to improve accuracy beyond the CLEAN model.

7.3.2 Performance of the denoising techniques

The second set of experiments evaluates performance of the DAE and DNCNN in noisy conditions. These tests use CLEAN and MATCH training data, and also examine how best to train the CNN classifier, on either clean data or vestigial data. Classification accuracy is measured across all four noise types and SNRs using the DAE and DNCNN methods trained on data matched to the specific noise type and SNR under test. Methods DAE-MATCH-CLEAN and DNCNN-MATCH-CLEAN output their denoised spectrogram features into a CNN classifier trained on clean data, while methods DAE-MATCH-VES and DNCNN-MATCH-VES output into a CNN classifier trained on vestigial data. Table 7.1 shows specific configuration details on these systems. For comparison, the performance of the clean trained CNN model (CLEAN) and matched CNN models (MATCH) are included with classification accuracies shown in Figure 7.3.

Figure 7.3 shows that the two denoising methods using the vestigial trained classifier (DAE-MATCH-VES and DNCNN-MATCH-VES) attain best performance and their accuracy is almost equal in all noise conditions. When these two denoising approaches are applied to the clean-trained classifier their performance reduces. This suggests that the denoising methods are not able to remove the contaminating noise completely. However, classifying the output spectrograms using a classifier trained on the vestigial noise is able to recover performance. The results also suggest that the DAE is better able to remove noise and minimise distortion as its mean performance using the clean-trained classifier is higher than the DNCNN with the clean classifier as shown in Table 7.1. Methods performing better when evaluated against the clean-trained classifier indicate that they match the original clean conditions more closely.

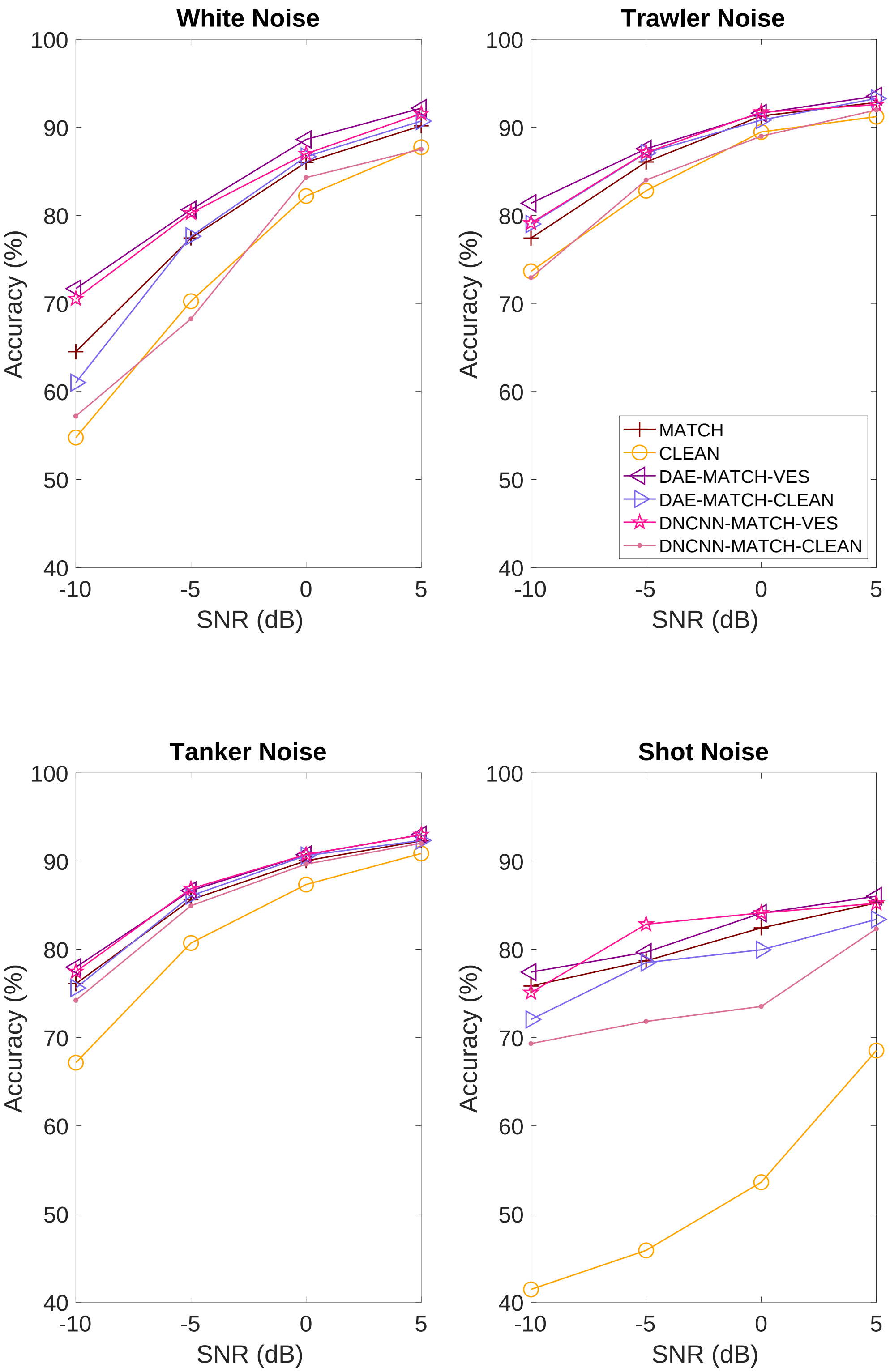
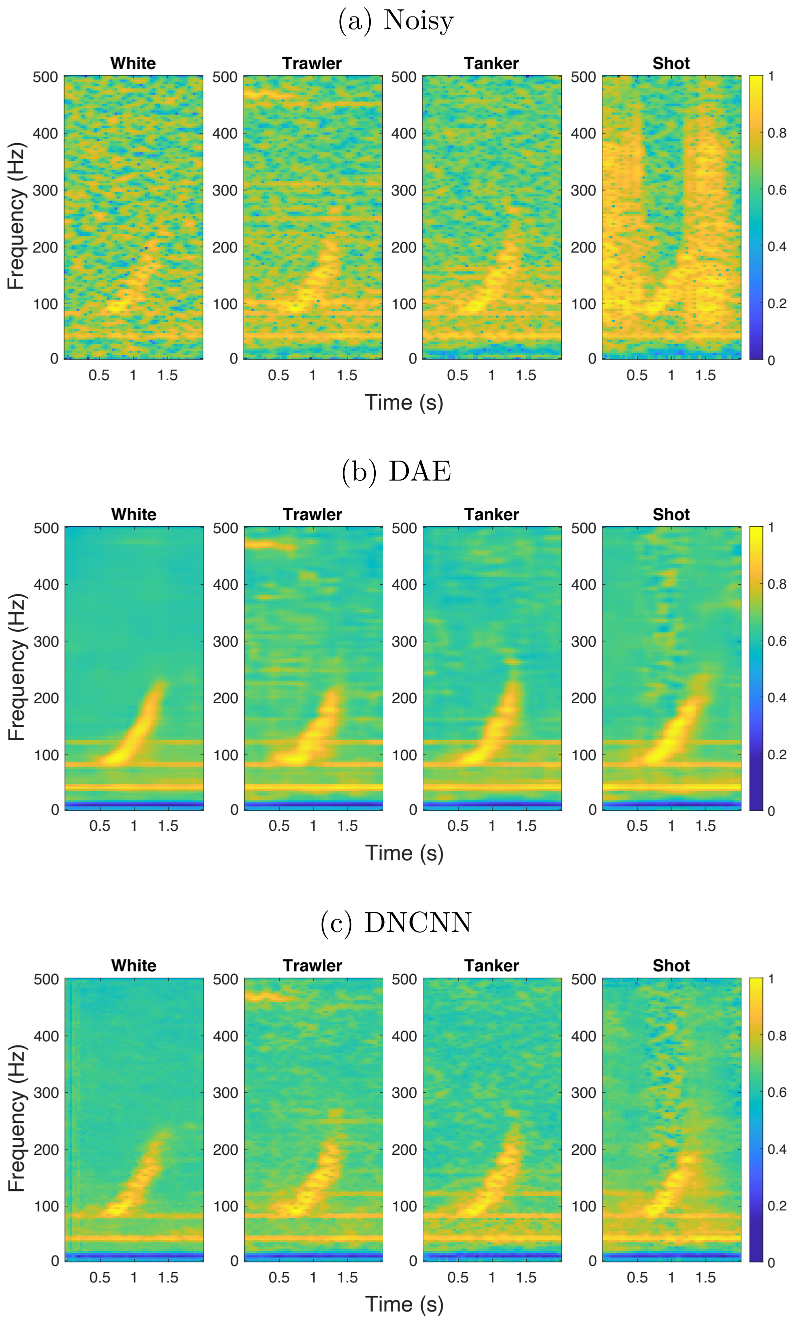


Fig. 7.3 NARW detection accuracies when applying the denoising autoencoder (DAE) and denoising CNN (DNCNN) to the four noise types at SNRs from -10dB to 5dB.


(a) Noisy

(b) DAE

(c) DNCNN

Fig. 7.4 (a) shows spectrograms of a single NARW upcall (as displayed in Figure 7.1) that has been contaminated with white, trawler, tanker and shot noises at an SNR of -5dB. Row (b) and (c) show the corresponding denoised spectrograms as produced by the DAE (b) and DNCNN (c) methods. The colourbar displays an amplitude range of 0 to 1 as these spectrograms are output from the denoising methods that are themselves trained on spectrograms with normalised energies.

To visually illustrate the denoising ability of the DAE and DNCNN, the top row of Figure 7.4 shows a single upcall example that has been contaminated by each of the four noise types at an SNR of -5dB. For comparison, the original noise-free upcall is shown in Figure 7.1. The bottom two rows show spectrograms resulting from denoising with the DAE and DNCNN, and all spectrograms are shown using the same amplitude scale. Figure 7.4 shows that slightly more vestigial components remain after the DNCNN has been applied which may explain its lower performance compared to the DAE in Table 7.1.

As a final investigation, the confusions between the three classes {*upcall(U)*, *gunshot(G)* and *not-NARW(NW)*} are examined across the four noise types. Tables 7.3 and 7.4 show confusion matrices for white noise and shot noise at an SNR of 0dB with no denoising (i.e. CLEAN). Confusions in tanker and trawler noises were very similar to those in white noise and so are not shown. In white noise, gunshots are classified more accurately than upcalls, while in shot noise, upcalls are classified more accurately. This is attributed to the shot noise having more similar characteristics to gunshot vocalisations and so introducing more confusion. Tables 7.5 and 7.6 show confusion matrices for the same two scenarios but now with the DAE applied (specifically DAE-MATCH-CLEAN). The primary effect of denoising in white noise is to reduce the percentage of not-NARW instances that are misclassified as either upcalls or gunshots, which represents a reduction in false alarms. This also happens when denoising in shot noise, but in addition, denoising also reduces the large number of gunshots that were misclassified as upcalls and are now classified correctly.

Table 7.3 Confusion matrix for no denoising on white noise at 0dB SNR.

	U	G	NW
U	76%	6%	18%
G	1%	89%	10%
NW	9%	9%	82%

Table 7.4 Confusion matrix for no denoising on shot noise at 0dB SNR.

	U	G	NW
U	58%	3%	39%
G	33%	51%	16%
NW	37%	21%	52%

Table 7.5 Confusion matrix for DAE on white noise at 0dB SNR.

	U	G	NW
U	75%	2%	23%
G	0%	89%	11%
NW	4%	1%	95%

Table 7.6 Confusion matrix for DAE on shot noise at 0dB SNR.

	U	G	NW
U	84%	0%	16%
G	0%	81%	19%
NW	24%	1%	75%

7.3.3 Denoising with augmented training

In previous tests the denoising method was trained on the noise condition under test. In this section, the denoising training is no longer matched to the noise condition under test and instead is trained on different noise and SNR conditions. Specifically, two scenarios are considered. First, where the denoiser is trained on all four noises and four SNRs (DAE-GENERIC-VES and DNCNN-GENERIC-VES) and secondly where training is on the three noise types that are not under test, which gives an unseen test condition (DAE-UNSEEN-VES and DNCNN-UNSEEN-VES). Given its superior performance in the previous section, all tests use the classifier trained on vestigial data rather than the clean-trained model. For comparison, results with no denoising are also shown and include the CLEAN model, GENERIC model and UNSEEN model, as defined in Table 7.1, with results shown in Figure 7.5.

Methods that include training across all noise types and SNRs (GENERIC, DAE-GENERIC-VES and DNCNN-GENERIC-VES) achieve highest accuracies across all test conditions. This is attributed to the models having been trained on noise data that

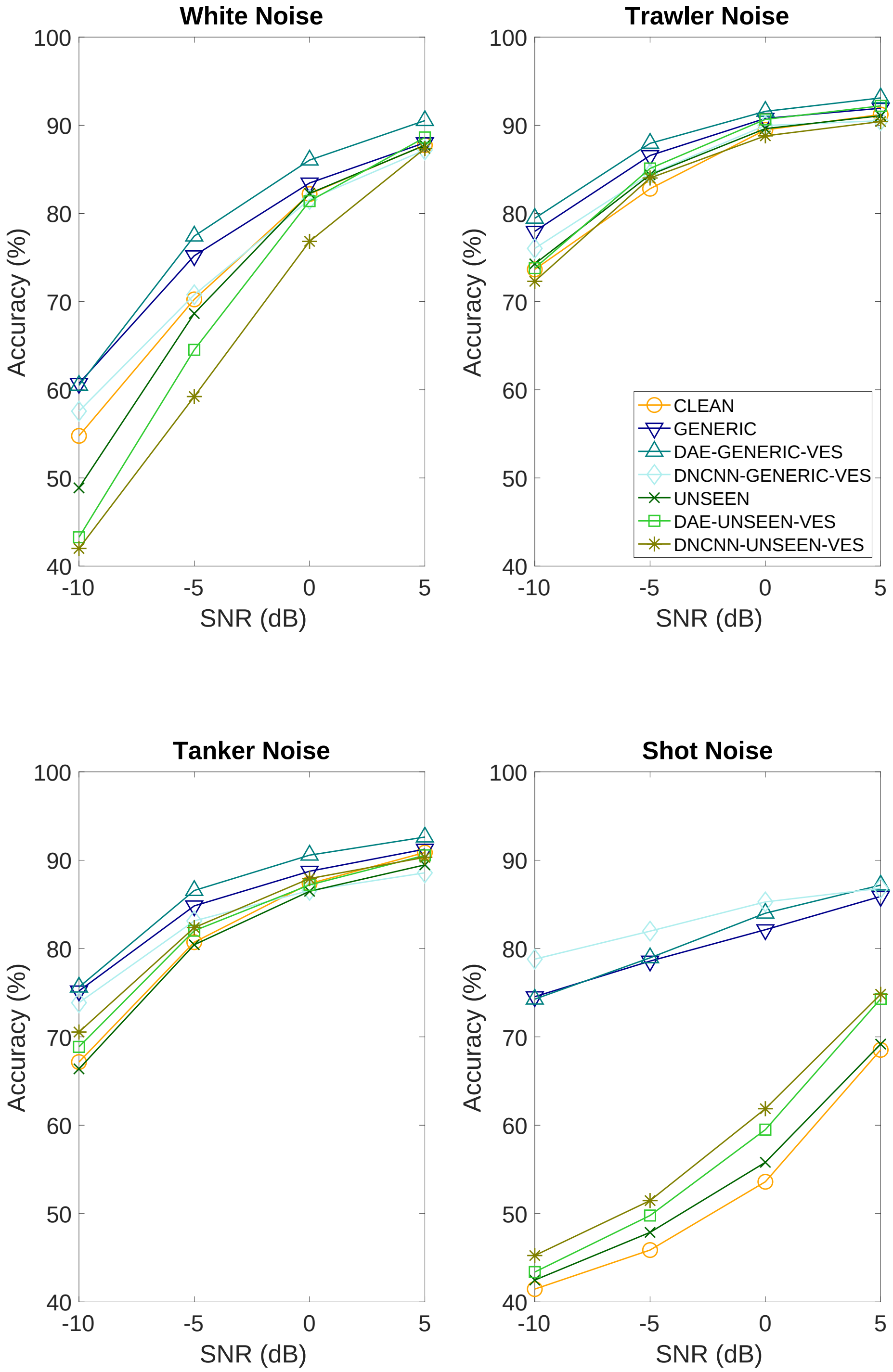
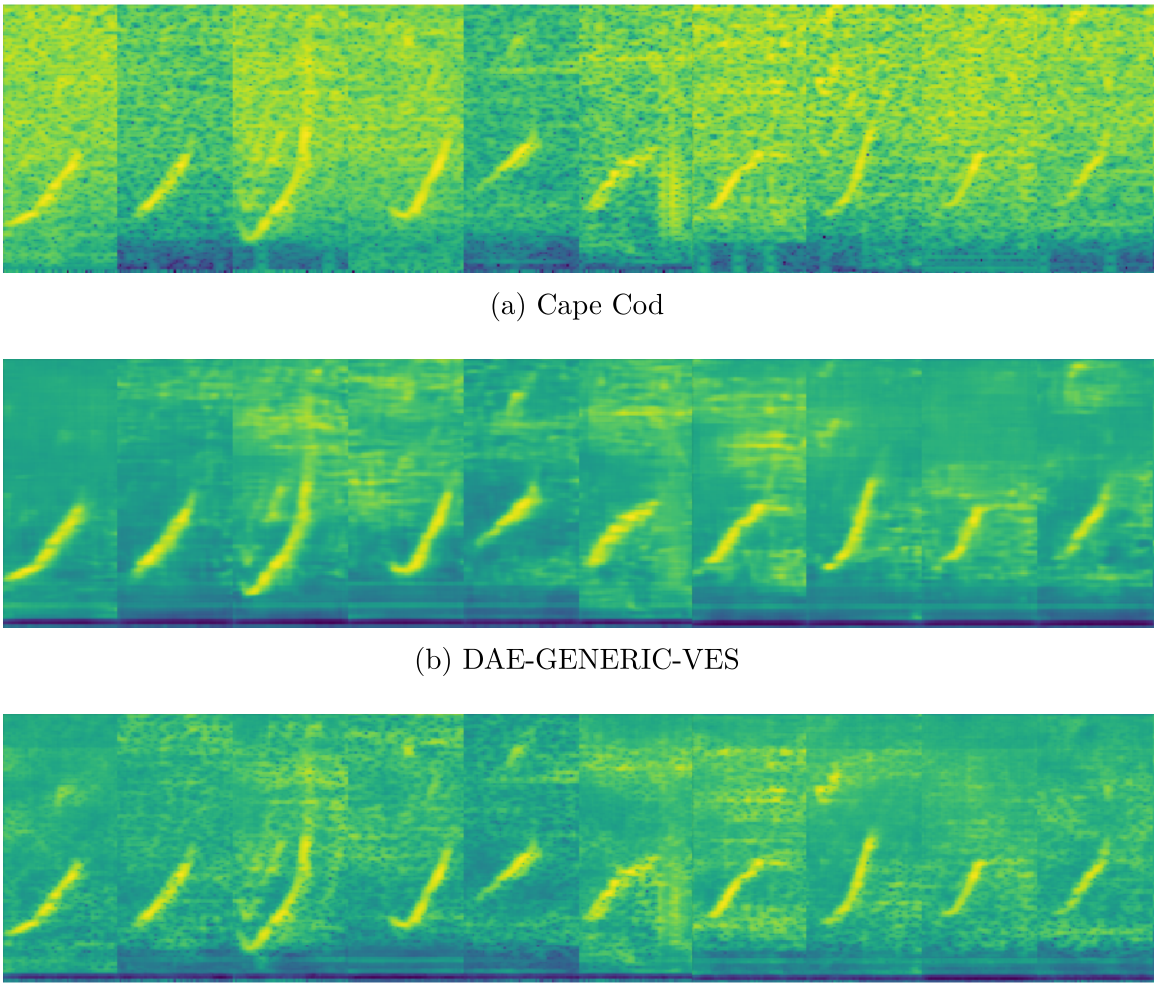


Fig. 7.5 NARW detection accuracies using denoising autoencoders and denoising CNNs when testing the four noise types at SNRs from -10dB to 5dB. Results are shown with the denoising methods trained generically on all noise types or on noises not used in testing.

has similar characteristics to the specific test condition, whether it be in the denoising process (DAE-GENERIC-VES and DNCNN-GENERIC-VES) or during classification (GENERIC). Moving to the unseen noise situations, where training does not include examples of the specific noise type under test, this leads to a reduction in accuracy for all systems (UNSEEN, DAE-UNSEEN-VES and DNCNN-UNSEEN-VES). Whilst testing in white noise and shot noise, accuracy falls substantially below that of the equivalent systems trained on all noise types (i.e. the GENERIC systems), while for tanker and trawler noises the reduction in performance is much less. This is attributed to the similarity between tanker and trawler noises which allows the methods to learn at least some characteristics of the unseen noise and thereby perform better than the clean-trained model. As the performance on tanker and trawler is largely maintained on the UNSEEN conditions this indicates generic denoising models are effective against similar noises.

7.4 Experimental evaluation in real noise conditions

The evaluation of the denoising methods in the previous sections used simulated noisy conditions by mixing clean audio with different noise types at varying SNRs. This approach is well suited for controlled evaluations of performance. An alternative scenario is now considered where the performance of NARW detection on real noisy data is investigated. For this evaluation, data is taken from the Cape Cod corpus which was described in Chapter 2.6.3 and collected from a marine environment different from the Stellwagen corpus. Tests now consider this as a detection problem as Cape Cod only contains two classes and therefore a matched two class Stellwagen corpus is used. Spectrogram analysis and listening to recordings has revealed them to contain significant amounts of different noise types which therefore represent a genuine unseen condition, not a simulated one. To illustrate the recordings from Cape Cod, Figure 7.6a shows ten example spectrograms of upcalls. They show that continuous broadband noise is present in most recordings as well as shorter duration impulses and some tonal noise, depending on the particular example.



(a) Cape Cod

(b) DAE-GENERIC-VES

(c) DNCNN-GENERIC-VES

Fig. 7.6 Ten example spectrograms taken from the Cape Cod test corpus are show in row (a). Row (b) displays the same spectrograms after denoising using the DAE-GENERIC-VES model. Row (c) displays the same spectrograms after denoising using the DNCNN-GENERIC-VES model.

Based on the evaluation in Section 7.3.3 in the unseen noise condition, performance of the Cape Cod data is now evaluated using the CLEAN, GENERIC, DAE-GENERIC-VES and DNCNN-GENERIC-VES configurations. Instead of measuring classification

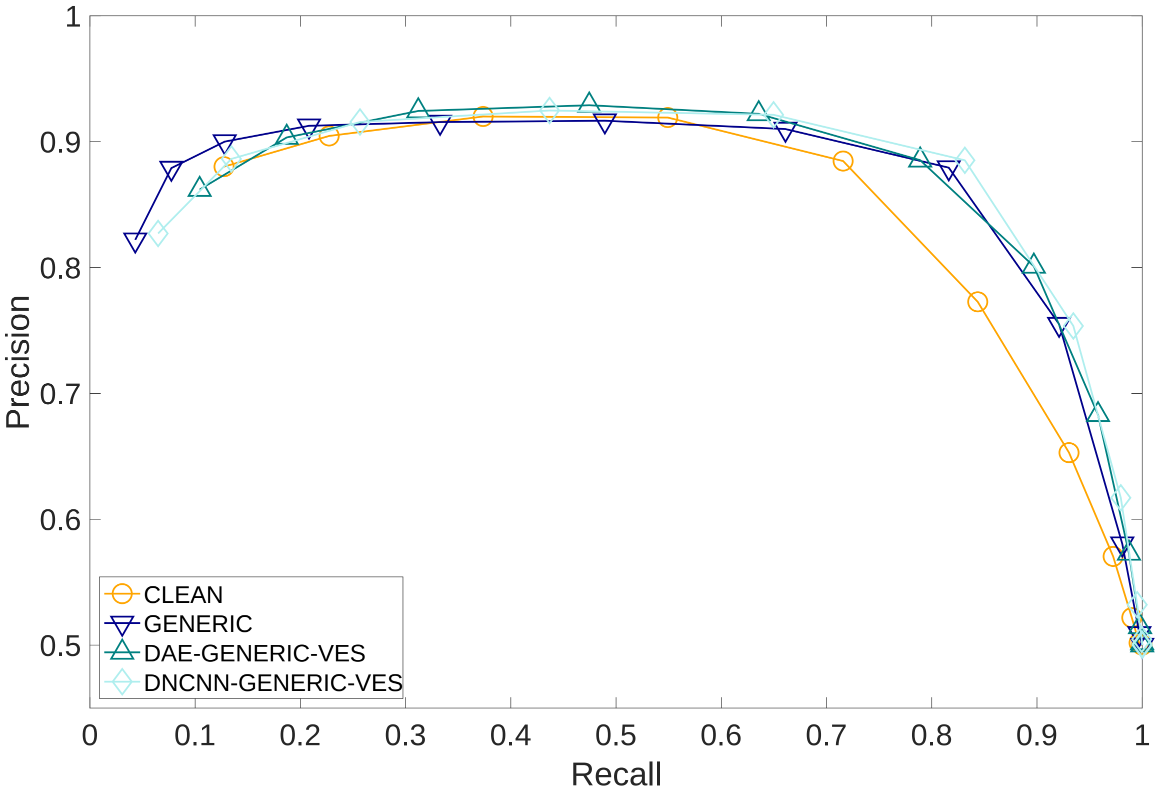


Fig. 7.7 Precision-recall curves for the CLEAN, GENERIC, DAE-GENERIC-VES and DNCNN-GENERIC-VES models that are trained on Stellwagen data and tested on unseen recordings from Cape Cod.

accuracy, as has been done previously, these tests consider the task of NARW detection (i.e. detecting whether a NARW is present or not in a recording). For a practical NARW detection system, knowing the precision and recall performance can be more useful than classification accuracy. Consequently, the system is evaluated using these metrics with results shown for the four systems as precision-recall curves in Figure 7.7. As previous tests used the three class Stellwagen corpus, this test trains new models using a two class Stellwagen corpus. This two class corpus is outlined in Chapter 2.6.3. The DAE-GENERIC-VES, DNCNN-GENERIC-VES and GENERIC systems have similar precision-recall profiles. All of the proposed systems outperform the CLEAN system, particularly at higher levels of recall, where their precision is substantially better. This is investigated further in Figure 7.6b & 7.6c which shows denoised spectrograms from DAE-GENERIC and DNCNN-GENERIC models. Both denoising methods appear visually to be effective at removing much of the noise present in the original spectrograms of Figure 7.6a however, both do leave some artefacts from denoising. This reinforces the benefit of using a classifier trained on the vestigial signal rather than on clean data. Overall the DAE-GENERIC-VES of Figure 7.6b appears to be slightly cleaner than the DNCNN-GENERIC-VES of Figure 7.6c. NARW upcalls do appear to be highlighted in both Figure 7.6b & 7.6c which could make this method of denoising beneficial when humans are manually processing PAM recordings.

To compare with the previous tests in Section 7.3, the classification accuracy was also measured for the four methods presented in Figure 7.7 and found that DAE-GENERIC-VES achieved the highest accuracy of 84.3%, followed by DNCNN-GENERIC-VES at 83.8%. The GENERIC system attained 81.7% and CLEAN 79.5%. From Table 7.1, the DAE-GENERIC-VES method also outperforms DNCNN-GENERIC-VES, and both improve over the CLEAN model. These results show the potential performance increase available when using the DAE to denoise spectrograms of NARW vocalisations.

Figure 7.7 also provides an indirect comparison to results published by Shiu et al. [188] where they produced a similar precision-recall curve to compare classifiers from the DCLDE 2013 conference challenge to their own implementation of a neural network classifier. The results presented in this work are not directly comparable as they have had noise artificially added and attempted denoising prior to augmented classification however, they do provide an indication of performance for the same dataset. It should be noted that the test data used in [188] is from the DCLDE challenge and was not available outside of the conference period. The test data used for Figure 7.7 is part of the Cape Cod dataset however comparison of precision-recall results still indicates the overall performance of the classifier. Figure 7.7 shows the highest precision-recall to be 0.90 precision and 0.84 recall. For [188] this is 0.80 precision and 0.89 recall. Overall each threshold step of the precision-recall plot delivers a close but slightly varied performance between this work and [188]. As Shiu et al. used non-corrupted data to train the model and achieved similar or lower precision and recall values, in comparison this work proves to be successful when dealing with noisy conditions.

7.4.1 Classification processing times

An important consideration when deploying a practical NARW detection system is the processing time required to make a decision. This is examined by measuring the time taken from receiving a two-second block of audio to making a classification decision, which includes computing the spectrogram, denoising (where applied) and classification. Times were computed by averaging timings for individual two-second blocks across the entire test set of recordings. The tests were performed on an Intel Quad Core i7 2.8GHz CPU which is a more realistic test than using a GPU, as was used in training. Three systems were evaluated: CLEAN, DAE-GENERIC-VES and DNCNN-GENERIC-VES, with the total time taken to process each two-second block broken down into the spectrogram extraction, denoising and classification times and shown in Table 7.7. This shows that all methods can process a two-second recording well within real-time constraints.

The slowest method was the DNCNN-GENERIC-VES, where the majority of processing is taken by the DNCNN although this is still capable of operating at 35-times real-time. The DAE-GENERIC-VES method of denoising was substantially faster, primarily due to the DAE denoising method operating eight times faster than the DNCNN denoising, which is due to it having fewer layers. Spectrogram extraction

Method	Spectrogram	Denoising	Classification	Total (ms)
CLEAN	0.72	-	2.63	3.35
DAE	0.72	6.40	2.63	9.75
DNCNN	0.72	53.02	2.63	56.37

Table 7.7 Mean processing times (in ms) for the spectrogram extraction, denoising and classification operations for the CLEAN, DAE and DNCNN methods when applied to a two-second audio recording.

is the fastest of all stages, requiring just 0.72ms. In a practical deployment, these very fast classification times would allow a single CPU to process multiple channels of hydrophone array data simultaneously in real-time, 205 channels for the DAE and 35 channels for the DNCNN, ignoring multiplexing overheads.

7.5 Summary & discussion

This chapter explored the suitability of the DAE and DNCNN methods of denoising in a range of noise conditions to consider more closely potential ocean noise environments. Tanker, trawler, shot, and white noise were used to evaluate the DAE and DNCNN in a series of more in-depth tests. Tests also considered the problem as a classification task with three classes, compared to a detection task as seen previously. The DAE and DNCNN were tested with a range of system configurations, detailed in Table 7.1, where augmented training, denoising and blind testing were all evaluated.

Tests found methods using augmentation outperformed those not augmenting training data. Similarly, testing using the DAE for pre-processing spectrogram images saw an increase in detection accuracy in all noise conditions and across all SNRs tested. Additional tests explored the effect of retraining the CNN classifier with vestigial data, compared to the use of a clean-trained CNN classifier. Results showed that retraining on the vestigial signal and using the DAE, provided the highest detection accuracy.

Overall the system configuration DAE-GENERIC-VES, which used augmented training for both the DAE and CNN classifier, and used the DAE to pre-process all of the spectrograms before retraining the CNN classifier on the vestigial signal, gave the highest accuracy across the largest range of simulated noise conditions and SNRs.

Finally, DAE-GENERIC-VES and DNCNN-GENERIC-VES configurations were both evaluated in a condition of natural noise using the Cape Cod corpus for testing and the Stellwagen corpus to train the denoising method and classifier. Denoising using a real-world unseen condition represents the closest experimental setup possible for mimicking a real-world scenario. When Cape Cod was used to test the clean Stellwagen model, a detection accuracy of 79.5% is attained. Using the DAE-GENERIC-VES configuration, detection accuracy improved to 84.3%, showing the real-world benefit of using the proposed configuration.

The results presented show the potential effectiveness of using the denoising autoencoder and advantage of using augmented data when training. Throughout testing, augmentation consistently provided an increase in accuracy over methods not using denoising or augmentation. It was also discovered that classification accuracy increased with a more varied set of augmented data. For example, using all noise types at all SNRs did not decrease accuracy and helped to improved performance in many noise conditions. It is suggested that the collection of a large array of ocean sounds could allow a completely generic DAE to be trained, where new noises could be successfully denoised without dedicated retraining on each new sound.

Measurement of processing times revealed the DAE to operate at 205 times real-time compared to 35 times real-time for the DNCNN. The faster operation and higher classification accuracy achieved by the DAE suggest this is a better choice for denoising within the domain of robust detection of NARWs.

Chapter 8

Unsupervised adaptation of classification models for new conditions

8.1 Introduction

This chapter uses unsupervised adaptation on current classification models with the aim of increasing accuracy when classifying NARW vocalisations in new noise and environmental conditions, originally unseen. Previous work in this thesis has focused on developing methods for enhancing, denoising, or augmenting current datasets for new unseen conditions and retraining the original model to improve accuracy. In contrast, this chapter uses adversarial discriminative domain adaptation (ADDA) [204] to update a current model to make it more suitable in new conditions or domains. As previously seen in Chapter 7, when a classifier processes PAM recordings from a new condition (normally a change in noise or environment), reported accuracy drops. For example in Chapter 7.4, performance from the proposed CNN system when trained and tested with the Stellwagen corpus was 97.91%. Using the same model, this dropped significantly to 79.5% when tested with the Cape Cod corpus. In ideal conditions, a new model matching the new domain could be trained and performance maximised. In real-world use, generating new models, and collecting and labelling data take time and resources that are not always available. For example, surveying a new area of ocean with different environment characteristics could cause a model trained on a previous domain to perform poorly and miss NARW vocalisations. Domain adaptation aims to adapt a current model to bridge the gap between a matched condition (where the model and test data are from the same domain) where accuracy is maximised and a mismatched condition (where the model and test data are not from the same domain) where accuracy has dropped.

The aim of this work is to consider scenarios where deployment conditions for NARW detection are changing and not necessarily matched to the source data used to train the underlying model [215]. A single adapted model, that can operate effectively under different operating conditions is proposed. In order to assess the baseline performance in both matched and unmatched conditions, a single environment and noise condition are used with the proposed CNN classifier developed in Chapter 3.3.4. Domain adaptation can then be explored to create a new model to restore performance in mismatched conditions, whilst still retaining equal performance in the original condition.

The remainder of the chapter is organised as follows. Section 8.2 provides a background on domain adaptation and areas where it has been previously used successfully. The application of ADDA is introduced in Section 8.3 with Section 8.3.1 exploring the process of adapting a model from one domain to another. Section 8.3.2 then details the evaluation carried out in Section 8.3.3 where results are presented in terms of a change in noise conditions and environmental conditions.

8.2 Background

Domain adaptation is an active area of research within deep learning [44] with many applications aiming to transfer knowledge from an existing labeled domain to a new domain without retraining [242]. A limitation of current deep learning approaches is that they normally require thousands of labelled examples to train a model accurately. However, models do not generalise well to new target domains, where input data is substantially different from the original source domain. Training a new model with data from a new domain or using augmentation and retraining, as shown in Chapter 7.3.1, is often the only solution to achieve similar performance. In many situations, when met with a new domain these approaches are not possible, normally due to a lack of time, resources, or labelled data. For example for when surveying for NARWs in a new location or with new noise corruptions present, the current model might perform poorly and miss crucial detections. Domain adaptation works to improve detection accuracy in new conditions without labelled data, using an unsupervised approach. In practice using domain adaptation could be crucial in situations where labelled data, or resources to retrain a model are not available. Once these resources become available and labels are produced, models can be retrained but domain adaptation could yield an increase in accuracy before this is possible. Figure 8.1 visually shows the benefit of using domain adaptation.

Since the early 2010s, the popularity of domain adaptation methods has risen significantly [242]. Multiple domain adaptation techniques currently exist [222] with variations in the methodologies used to adapt domains. This chapter focuses on adversarial discriminative domain adaptation (ADDA) which encourages domain confusion within the model to allow target data to be classified more accurately [222].

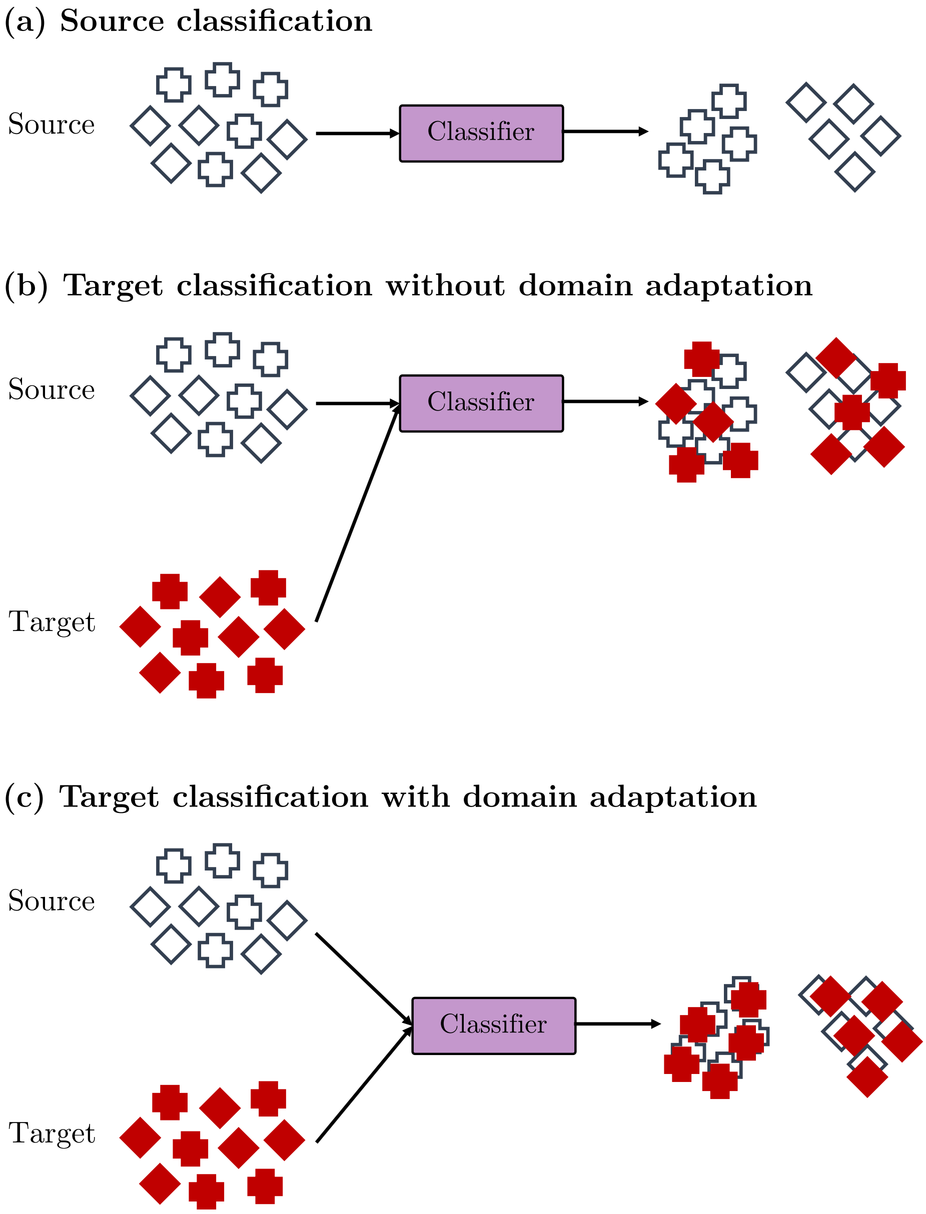

(a) Source classification
Source
Classifier
(b) Target classification without domain adaptation
Source
Classifier
Target
(c) Target classification with domain adaptation
Source
Classifier
Target

Fig. 8.1 A visual representation of why classification using domain adaptation can be beneficial. a) shows the original classification system. b) classification using the original system now with a new target domain, causing confused in target predictions. c) shows classification after the classifier has be domain adapted with better performance for the target domain.

ADDA has worked effectively in a range of image classification tasks, such as increasing accuracy of facial recognition when new images are presented from a different domain [132]. Gabriela Csurka also investigated ADDA, surveying a range domain adaptations problems and found successful domain adaptation solutions present in a variety of applications such as object detection, object recognition, speech recognition and sentiment analysis [44]. ADDA is now presented for the task of adapting domains when detecting NARWs.

When using a model previously trained on a source domain, M_S, to predict samples from a target domain, model performance generally degrades due to domain shift or dataset bias [242]. Domain shift is a change in the distribution of the model when compared to the distribution of the test scenario [196]. This occurs when related data is tested on a high performing model from an alternate domain. In the problem of NARW detection, two domain shift scenarios are considered, i) noise, ii) environment. A change in noise could occur due to the current noise level increasing or decreasing or when a new noise source is present. A change in environment could occur when surveying in a new location where ambient conditions are different from the source location. Dataset bias also can lead to in a drop in accuracy when a new domain is met. Bias within a dataset occurs when the dataset variance is low and generalisation between the samples is minimal. In a laboratory environment this could have a large effect when data collection is controlled and recorded samples are extremely similar. For NARW detection this is less likely, as ocean conditions and NARW vocalisations are rarely identical.

8.3 Application of domain adaptation for detection in changing conditions

This section now investigates the use of domain adaptation for improving detection accuracy of NARW vocalisations in changing conditions. First, Section 8.3.1 describes the process of ADDA and how it aims to combine multiple domain distributions. Second, Section 8.3.2 outlines the experimental approach and setup. Finally, an evaluation of results in multiple domains is provided in Section 8.3.3. Tests evaluate a change in conditions for noise level, environment, and both noise level and environment.

8.3.1 Implementation of adversarial discriminative domain adaptation

For implementation of ADDA it is assumed that target training data is available but without any labels which makes this method well suited to a new, unknown operating condition. Implementation of ADDA is a three-stage procedure which is shown in Figure 8.2. The first stage uses only the source data and associated class labels to train a CNN encoder, M_S, and classifier, C. This is the same procedure used to create the proposed CNN from Chapter 3.3.4 and is shown in Figure 8.2a. Figure 8.2a simply represents a separation of the CNN encoder from the classifier.

The second stage creates a target encoder, M_T, that aims to transform the target data into the same feature space as the source data and is illustrated in Figure 8.2b. This approach enables the same classifier, C, to be used for NARW detection for both the source and target data. The target encoder, M_T, is initialised using the weights from the source encoder, M_S.

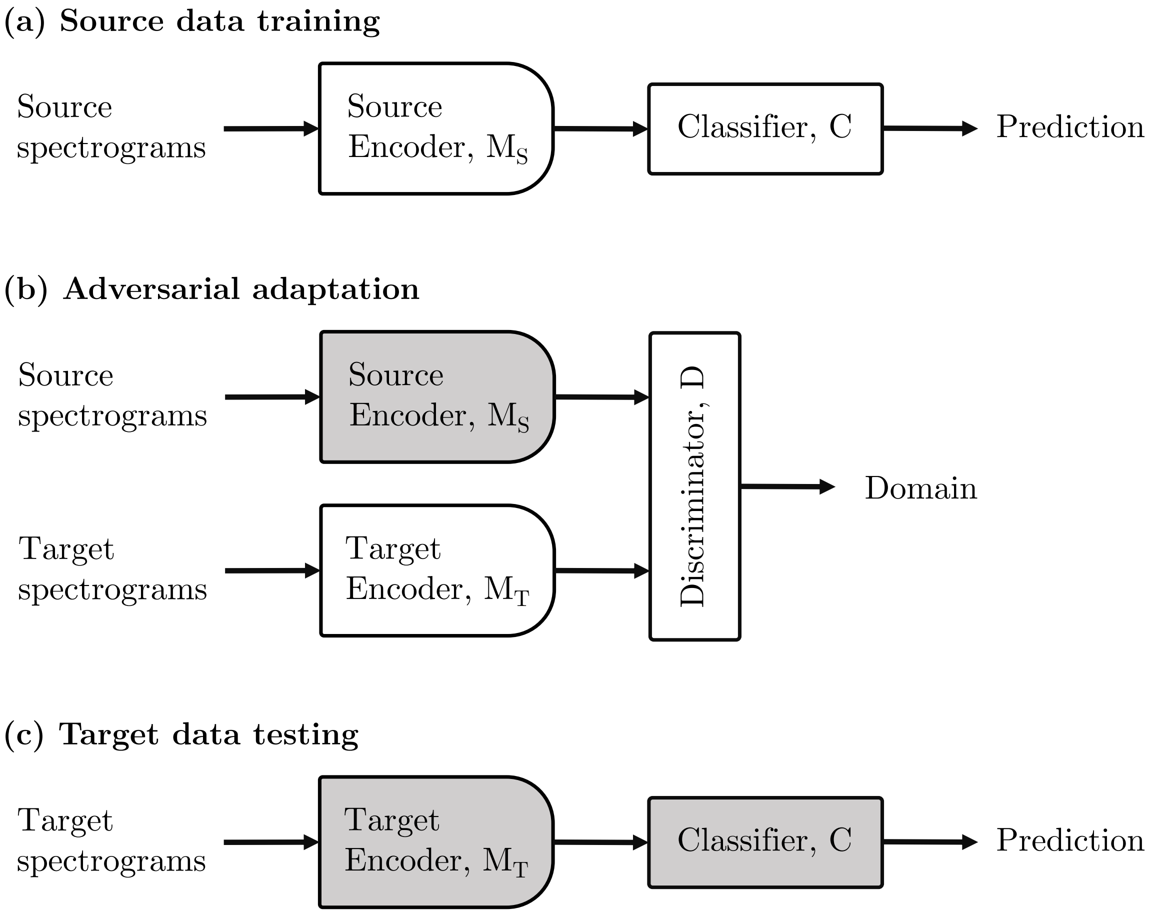


Fig. 8.2 Method of adversarial discriminative domain adaptation (ADDA) applied to spectrogram-based NARW detection. Gray boxes indicate a network that is fixed during training.

ADDA uses an adversarial loss to encourage the target encoder distribution to match the source encoder distribution. This is achieved by using a discriminator network to separate the domains. The discriminator, D, as shown in Figure 8.2b, is trained to differentiate the source domain and the target domain. The discriminator takes in encoded spectrogram features as input (from both M_S and M_T) and predicts the domain that each feature originally belonged too.

The discriminator then uses a loss to provide feedback to the layer weights during training to learn the differences between source and target domains. Similarly to the proposed CNN presented in Chapter 3.3.4, a loss is created to also train the target encoder, M_T, at the same time. However, unlike previous approaches, the loss to update M_T weights is calculated using the predictions from D for the target data, and an inverse of the target labels passed to D. Providing the inverse of the original target labels to the loss function will produce a loss between the source domain labels and the prediction from D of the target data. Theoretically, this loss should get larger as D gets better at separating the domains. Consequently, as training continues and optimisation of M_T updates weights based on a large loss value, the distribution of M_T will shift closer to that of the source domain. Eventually, predictions from D should tend to the source domain as both encoder distributions become similar. Subsequently, the loss for M_T will also be minimised and training can finish.

The third stage is shown in Figure 8.2c where testing of the target samples is carried out. Spectrograms from the target domain are transformed by the target encoder, M_T, into the source domain space. The classifier, C, then determines whether or not a NARW vocalisation is present. C is consistent for both the source and target domains, as adversarial training aims to shift the target domain into the same space as the source domain. ADDA is crucially an unsupervised method of adaptation and does not require a labelled target domain. This makes it suitable for use in new conditions where labels are not available.

For the evaluation of ADDA in Section 8.3.3 the neural network architectures for, M_S, M_T and C are consistent with those developed in Chapter 3.3.4 as they achieved the highest detection accuracy in tests. The discriminator, D uses a 3 layer fully connected network with 200, 100 & 1 node respectively, with the final layer used for generating predictions employing a sigmoid function for activation. All models used the Adam optimiser and trained for 200 epochs to ensure sufficient time for domain shift to occur.

8.3.2 Experimental setup

Experiments presented in Section 8.3.3 first examine how effective domain adaptation is at improving NARW detection in new operating environments. Second, their effect in new noise conditions is examined. Third, the effect of changing both the environment and noise is examined. The Cornell corpus, as described in Chapter 2.6.3, is used as the source dataset throughout testing, unless specifically signified. All tests that evaluate domain adaptation in Section 8.3.3 use a range of adaptation samples to assess performance in different situations. This directly relates to real-world use, where potentially there are only a limited number of a new domain samples available for adaptation. Tests use 10, 100, 1,000, & 10,000 adaptation samples to analyse the most suitable amount of adaptation data.

Environment

To examine the effect of changing environment, the Cape Cod corpus is used as a target environment. The Cape Cod corpus, detailed in Chapter 2.6.3, represents a different location to that of the Cornell corpus and therefore domain adaptation is used to maximise performance when only a Cornell-trained model is available for detection. In Section 8.3.3, multiple baseline accuracies are first recorded prior to domain adaptation. Firstly, the Cornell test data is evaluated against the Cornell model to provide detection accuracy in ideal conditions. Next, the same test is carried out for Cape Cod, with test and training data matched to the Cape Cod corpus. This provides the maximum performance for the Cape Cod environment. A mismatched condition is evaluated next with the Cape Cod test set used to evaluate performance of the Cornell model in conditions where the environment is not matched. Finally, domain adaptation is tested with a range of adaptation samples. The adaptation test uses the Cornell-trained model as the source domain and Cape Cod as the target domain.

Noise

The robustness of the proposed CNN detection system to changing noise conditions is now examined. As previously discussed, many sources contribute to sub-sea noise and thereby reduce the received signal-to-noise ratio of NARW vocalisations. Furthermore, sounds recorded from more distant NARWs will also be received with lower SNRs. To simulate noisy conditions white noise is used to corrupt the original audio. Specifically, white noise at an SNR of 0dB is added to the test samples of the Cornell corpus as this was previously found to have a significant impact on accuracy, and produce an alternate domain. Section 8.3.3 considers a range of scenarios to assess performance of the domain adaptation method. Firstly, the clean Cornell test data is evaluated against the clean Cornell model to provide detection accuracy in ideal clean conditions. Next, matched conditions are again evaluated but instead with noisy Cornell test data, against a noisy Cornell model. This provides the maximum performance in noisy conditions for the Cornell dataset. Detection accuracy in mismatched conditions is evaluated next with the noisy Cornell test set evaluated against the clean Cornell model. Finally, domain adaptation is tested with a range of samples available during training. The adaptation test uses the clean-trained Cornell model as the source domain and noisy Cornell as the target domain.

The third test condition that aims to represent both a change in the noise domain and environment domain, follows the same evaluation structure as described above and is presented last in Section 8.3.3.

8.3.3 Experimental results

An evaluation of tests described in Section 8.3.2 is now presented. Domain adaptation for improving the robustness of NARW detection is evaluated for three real-world scenarios, i) a change in environment, ii) a change in noise level, iii) a change in both environment and noise level. Domain adaptation tests are performed using 10, 100, 1,000 & 10,000 adaptation samples.

Changing environment

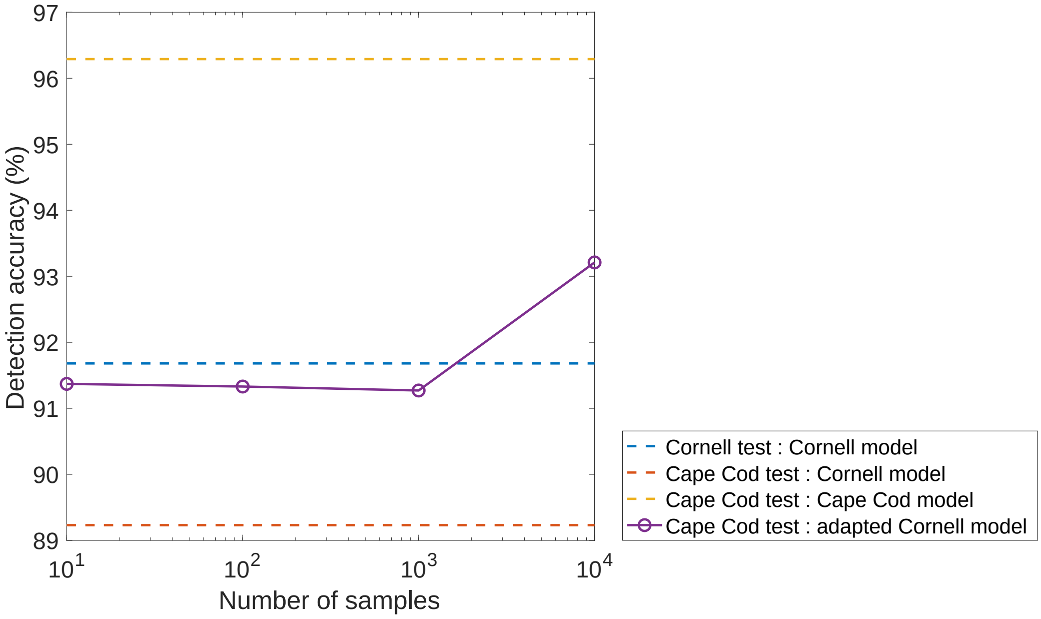


Fig. 8.3 Detection accuracy of the unsupervised adapted model as the number of Cape Cod samples is increased. All models start with a baseline of 10,000 samples from the Cornell set.

Tests in Figure 8.3 consider the use of domain adaptation when the current environment is different from the environment that was used to originally train the classification model. In this scenario the Cornell-trained model forms the source domain and increasing amounts of Cape Cod training data are used to create a new target domain that is tested on the Cornell model. Initially three baselines detection accuracies are recorded. When tested with Cornell data in ideal matched conditions the Cornell model achieves a detection accuracy of 91.7%. This drops to 89.23% when the target domain is tested against the same Cornell model. When using ADDA, as shown in Figure 8.3 performs improves over the mismatched condition when adaptation is carried out with as little as 10 samples from the target domain. A further improvement in accuracy can be found when using the maximum 10,000 samples during adaptation training, with accuracy increasing to 93.21%, a 4% improvement over the mismatched condition.

Changing noise

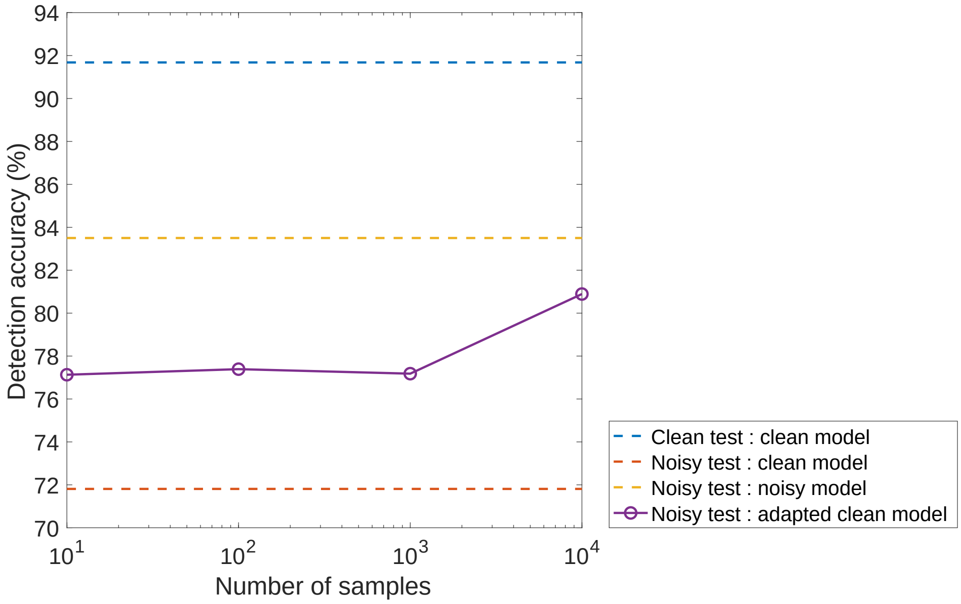


Fig. 8.4 Detection accuracy of the unsupervised adapted model as the number of noisy samples is increased. These test use a baseline Cornell model trained on 10,000 samples, before adaptation is applied.

Tests now consider unsupervised domain adaptation when the noise level has changed. White noise was added to the Cornell corpus at an SNR of 0dB to simulate this. As previous tests established, testing with a matched clean Cornell model achieves a detection accuracy of 91.7%, which decreases to 71.81% when tested in noisy conditions on the same clean-trained Cornell model. Figure 8.4 shows, in matched noisy conditions the detection accuracy sits roughly, equal distance between the mismatch and clean matched performance.

Varying amounts of the noisy Cornell training data are used to adapt the clean-trained Cornell source model. Figure 8.4 shows detection accuracy as the number of target samples is increased from 10 to 10,000. With a relatively small number of target samples, accuracy is increased from 71.81% with no adaptation to 77.13%. Using the maximum number of 10,000 target samples, accuracy increases to 80.9%, which provides a 9% increase in detection accuracy when the model is adapted.

Changing environment and noise

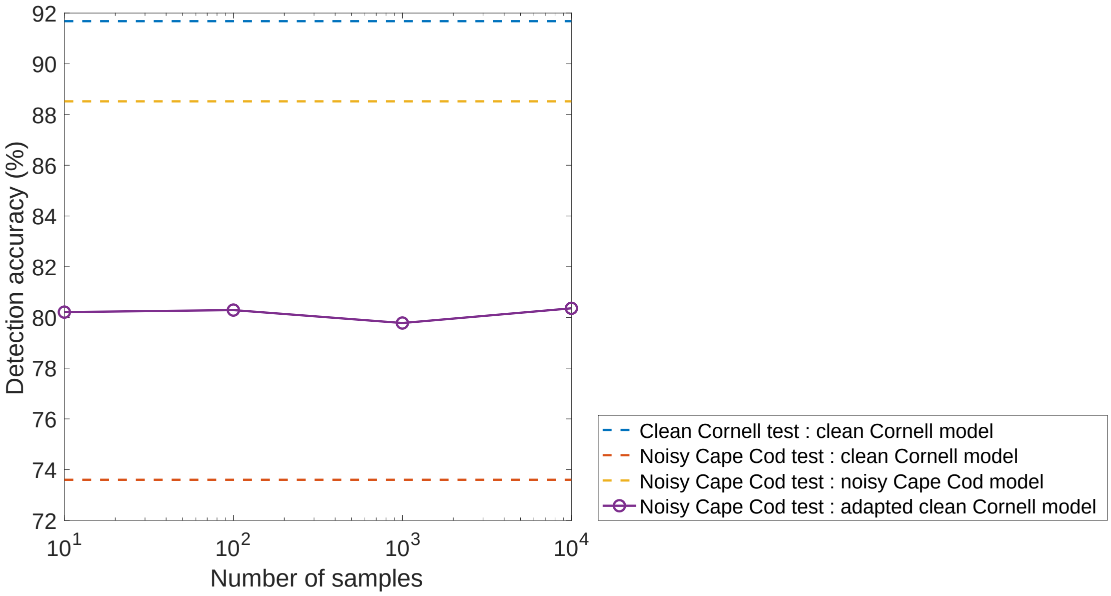


Fig. 8.5 Detection accuracy of the unsupervised adapted model as the number of noisy Cape Cod samples is increased.

As a final test, changes to both the environment and noise conditions are considered together. Target data is created by combining white noise at 0dB with the Cape Cod data. Figure 8.5 shows the detection accuracy of noisy Cape Cod data when tested against the adapted clean-trained Cornell model. In either matched clean conditions for Cornell, or matched noisy conditions for Cape Cod, performance is relatively strong, achieving 91.7% or 88.52% respectively. When a mismatch occurs, detection accuracy falls to 73.60%, indicating a large shift in domain between the test set and the underlying model. As Figure 8.5 shows, when using ADDA to adapt the source model to the target domain, an initial increase to 80.21% with as little as 10 target samples is seen. However, further increases in adaptation data give no increase in accuracy. This is attributed to difficulties in creating suitably stable models.

8.4 Summary & discussion

This chapter has investigated the use of domain adaptation for adapting a series of new target conditions to an original source condition to create a robust NARW detection system. The Cornell corpus has been used to represent an original location and fixed noise profile for which a corrupted noisy version and secondary corpus, have been used to test performance in changing conditions. When detecting NARWs a range of domain environments may be met that have not been previously included in classification models. In scenarios such as this, model adaptation may provide the highest increase in performance and enable the detection of NARWs that could be otherwise missed.

This investigation has shown that unsupervised adaptation using adversarial discriminative domain adaptation, is able to improve mismatched detection accuracy when the operating conditions change. Accuracy was increased in all conditions from the mismatched performance, with a maximum increase of 9% seen when a noisy target domain was evaluated. As this method of domain adaptation is unsupervised it does not require labelled target samples and therefore provides a realistic solution to domain shift and dataset bias in real-world conditions.

This chapter has considered one of the many domain adaptation methods [222], however further work in this area could evaluate alternative methods against ADDA to ensure the most suitable adaptation method for NARW detection is used. Although higher detection accuracies can be achieved when augmenting or training a new model for detection. ADDA provides a solution when alternative methods are not appropriate. Test results have shown domain adaptation to be a valuable addition to the proposed NARW detection system and the increase in detection accuracies indicate the value that it can add.

Chapter 9

Conclusion and Future Work

9.1 Overview

This thesis has presented methods to automatically detect the presence of NARW in a range of ocean conditions. A variety of machine learning methods are presented in Chapter 3 with the aim of producing a system capable of achieving high detection accuracies for use in real-world conditions. Chapter 3 also develops a range of deep learning algorithms with optimised performance for NARW detection and are subsequently benchmarked against other industry-standard machine learning techniques to indicate which methods are best for NARW detection. Chapter 4 introduces the problem of noise, providing a comparison of classical noise reduction techniques, both audio and image based, applied to noisy ocean recordings. Chapters 5 & 6 then build upon Chapter 4 but instead develop two differing deep learning methods of noise reduction aimed at producing a multi-noise robust NARW detector. A range of simulated noise conditions are presented in Chapter 7 with both the DAE and DNCNN processing pipelines further developed, to provide a robust NARW classification solution in all noise conditions. Further work in Chapter 8 explored unsupervised domain adaptation with the aim of making deep learning models more robust when augmentation or retraining were unsuitable.

9.2 Thesis conclusion

This thesis has been predominantly concerned with improving the automated detection of North Atlantic right whales from an acoustic source. Research in this area has been continuously gaining momentum, with the rise of recent NARW deaths [62] and the growing threat of extinction. Current methods of detection can be slow, manual, time consuming and require specific expertise. The development of an automated process aims to give experts more time for less tedious tasks whilst providing a potentially more accurate, real-time detector that can operate continuously. This thesis also aims to provide evidence for the suitability of using the developed detection system with low-powered, low-cost hardware that can run on-board small ocean-deployed platforms such as buoys or autonomous surface vehicles. Although in its infancy, the proposition of utilising ASVs to monitor the ocean for NARWs could provide further insights into their location and movement patterns and could mitigate potential threats. Theoretically, these factors should directly correlate to reducing NARW deaths and enable a sustainable increase in population.

The development of new deep learning architectures along with a comparison against traditional algorithms has shown CNNs to provide the best detection accuracy when processing ocean recordings containing NARW vocalisations. Analysis has shown that spectrogram features, presented as images to the CNN, produce the highest number of correctly detected upcall segments (92.62%). In comparison, detection accuracy from the time domain signal was substantially lower (70.85%). It is therefore recommended that detection uses spectrogram features of acoustic recordings when monitoring. Although this work found CNNs to perform best, alternative classifiers such as transformers, or time domain deep learning classifiers such as inception time, were not evaluated and have the potential to provide high accuracies. Further work could review additional classification methods for this application.

Investigations exploring noise within marine environments are largely concerned with the affect that anthropogenic sounds have on marine wildlife [42]. Little research has been carried out in regards to the best way of dealing with noise when passive acoustically monitoring. Chapter 4 provides development of noise reduction processing for NARW detection. Tests found that log features produce the highest accuracy in non-noisy conditions, with log histogram equalisation performing best in noisy conditions. Denoising deep learning models are developed in Chapters 5 & 6 to further analyse noise reduction, with tests evaluating a wider range of noise corruptions in Chapter 7. Real-world noises were synthetically added to the original dataset to provide a range of real life use cases. Denoising using the DAE or DNCNN outperformed the traditional methods investigated, with results demonstrating that the deep learning models provide the largest increase in detection accuracy, with accuracies after denoising approaching those observed in clean, non-noisy, conditions.

Finally, naturally noisy test data was evaluated to assess performance against non-synthetically altered recordings. The DAE and DNCNN were trained on noises not present within the naturally noisy test set. Both denoising methods were successful with detection accuracy improving in all cases. The DAE provided the largest improvement with accuracy increasing from 79.5% to 84.3%. These tests indicate that if a large, diverse collection of representative noises were used to train the denoiser, then denoising in all conditions could improved accuracy. Processing requirement analysis was also carried out and found the DAE to be more efficient than the DNCNN, however both methods leave sufficient overhead to process in real-time using low-powered and low-cost hardware. It is therefore expected that the proposed detection system would meet requirements for embedding on-board an ASV or static buoy.

In Chapter 8, an unsupervised method of adapting underlying CNN models was investigated. In contrast to other techniques explored within this work, Chapter 8 explores domain adaptation which aims to shift encoder distributions of new environments or domains, to match that of a current distribution and therefore improve accuracy. The evaluation discovered domain adaptation to be successful in a range of conditions and a valuable addition to a NARW detection system for scenarios where current and new conditions are significantly varied.

In summary, it is recommended that the developed CNN architecture be used for detection of NARW vocalisations with a pre-processing DAE layer to reduce noise within recordings. The DAE should be trained on a large mixed corpus of representative ocean sounds, as testing has shown that generic models work as well as noise specific models. Finally, training the CNN with denoised vestigial segments is recommended as it has proven to provide the highest detection accuracy. The addition of domain adaptation would further improve performance when accuracy drops due to a mismatch in PAM conditions.

9.3 Future work

Some of the possible avenues for future work that would compliment this thesis are now discussed.

9.3.1 Real-world usage

Whilst consider future avenues for experimentation is also important to consider how this work can be applied to real-world situations. Much of the current experimentation does not focus on a real-world situation nor discusses the practise of detecting NARWs in the field. As stated in Chapter 2.6.1, NARWs are known to infrequently produce upcalls and therefore consideration of the weighting of classes within this classification problem should be assessed further. In the tests presented, classes were equally divided however previous research [133] shows that this is unlikely to occur and weighting towards *"not-NARW"* should be addressed as to not receive an overwhelming number of false alarms. When using a more natural weighting of classes, the results could be evaluated in regards to a real-world use case, for example using a more useful metric such as an analyst reviewing detection events per hour [188]. If a classification system can report less false alarms than a human operator can manage per hour, then confidence that the system can be used in a laboratory setting can be established.

9.3.2 Generic cetacean classification

Whilst researching cetaceans it has become increasingly clear that availability to access high quality datasets is incredibly low. High quality in this instance describes accurately labelled, consistently organised and well documented recordings of cetaceans with as many repetitive detected vocalisations as possible. Within other fields, datasets are more widely available either through community distribution [9] or by manual data collection. Since the latter can be difficult, expensive and provides potential disruption to cetaceans, it is more important than ever to implement channels capable of sharing data across all researchers. Overall, the implementation of static buoys and autonomous surface vehicles to carry out passive acoustic monitoring provides the best platform for NARW detection as interference with NARWs is minimal whilst enabling continuous monitoring. Since reliance on these platforms is growing, the methods to accurately automate detection (such as the NARW detection system proposed in Chapter 3.3.4) are becoming fundamental to the success of mitigation projects such as the NOAAs NARW recovery plan [62].

A large challenge when developing machine learning models is obtaining access to suitable datasets. Without suitable datasets that provide accurate ground truths further experiments can be unstable or produce less than satisfactory results. Many algorithms, such as those used for deep learning, produce better detection accuracies with larger amounts of training data. The problem of cetacean detection is not only relevant to NARWs and therefore producing similar systems for the detection of other species would enable mitigation and research in the same way that it has for NARWs. In an ideal scenario a single system would be capable of producing multi-species classifications. Using a single classifier to detect more than one marine mammal would be a significant step in this direction. As previously discussed in Chapter 2.2 many baleen whales produce similar frequency vocalisations and therefore could be suitable candidates for expansion of this proposed NARW detection system. Producing a multi-species classification system would enable multiple species to be monitored using one stream of acoustic data and one processing pipeline, in contrast to n number of systems running over the same data for each species of interest. This would present as a potentially dramatic reduction in processing costs and detection hardware.

9.3.3 Noise collection

In Chapters 4, 5, 6 & 7 background noise has been used to simulate a noisy ocean environment with the aim of creating a more robust detector than one trained in only clean conditions. Chapter 4 first introduces a single white noise type that was intended to represent a range of noise corruptions across the frequency spectrum. Further noises introduced in Chapter 7 were taken from ocean recordings of shipping vessels, fishing trawlers and a shot noise database. Tests in Chapter 7 discovered that generic noisy datasets were able to perform better across all noise types when training data was originally augmented with instances of those sounds. In order to produce a model robust to a wider range of noise corruptions it is concluded that gathering more ocean representative sounds and augmenting training data would provide the best approach, potentially capable of accurately detecting NARWs in a range of noise environments. Furthermore, tests in Chapter 7 also saw that similar noise corruption can bolster performance when a similar, but not identical noise is under test in a new environment. Tests to determine the extent of this finding would be informative, as a generic model may be attainable once enough noise examples were augmenting the training data. Noise recordings, made independently of cetacean detections are also of great importance when building a robust NARW detection system. As well as creating a space for sharing labelled cetacean recordings, emphasis should also be applied to a wide range of ocean sounds that can corrupt the clarity of cetacean vocalisations. Building a larger catalogue of ocean sounds could provide the ability to generate the highest performing noise-robust models.

9.3.4 Autonomous vehicle trials

The use of autonomous surface vehicles has been discussed in Chapter 2.4.2 where, after analysis of various data collection platforms, ASVs were found to provide a balance between cost and the ability to continuously survey regions of ocean without being a potential hazard to marine wildlife. The use of ASVs in real-world tests is reasonably limited as minor hardware issues plague widespread use [16]. Future work would benefit from deploying hardware payloads into the hulls of ASVs and monitoring performance from the on-board communication and detection systems. Time-to-notify, the time taken to detect a NARW and receive a notification of detection could be observed during testing using local and remote timestamps. Although final tests in Chapter 7.4.1 have shown the operation of the detection system to run in real-time with overhead for concurrent operations, performance in the real-world scenario has not been assessed. Running tests using accurate hardware and *in situ* conditions should provide reliable metrics as to the real-world suitability of the full detection system.

Bibliography

[1] Marwa A Abd El-Fattah et al. “Speech enhancement with an adaptive Wiener filter”. In: *International Journal of Speech Technology* 17.1 (2014), pp. 53–64.

[2] Abdelrahman Abdelhamed, Stephen Lin, and Michael S. Brown. “A High-Quality Denoising Dataset for Smartphone Cameras”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* June 2018.

[3] Caroline R. et al. “Comparative Review of the Regional Marine Mammal Mitigation Guidelines Implemented During Industrial Seismic Surveys, and Guidance Towards a Worldwide Standard”. In: *Journal of International Wildlife Law & Policy* 10.1 (2007), pp. 1–27. DOI: 10.1080/13880290701229838.

[4] Mark V Albert et al. “Fall classification by machine learning using mobile phones”. In: *PloS one* 7.5 (2012), e36556.

[5] Shahin Amiriparian et al. *Sequence to sequence autoencoders for unsupervised representation learning from audio.* Universität Augsburg, 2017.

[6] Anand Atreya and Daniel O’Shea. “Novel lossy compression algorithms with stacked autoencoders”. In: *Tech. Rep.* (2009).

[7] Oluwaseyi P Babalola et al. “Detection of Bryde’s whale short pulse calls using time domain features with hidden Markov models”. In: *SAIEE Africa Research Journal* 112.1 (2021), pp. 15–23.

[8] Anthony Bagnall et al. “The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances”. In: *Data Mining and Knowledge Discovery* 31.3 (2017), pp. 606–660.

[9] Anthony Bagnall et al. *Time Series Classification Dataset Listing.* URL: http://www.timeseriesclassification.com/dataset.php.

[10] John Bannister et al. “Status of southern right whales (Eubalaena australis) off Australia”. In: *J. Cetacean Res. Manage.* (2020), pp. 103–110.

[11] Jay Barlow. “The abundance of cetaceans in California waters. Part I: Ship surveys in summer and fall of 1991”. In: *Oceanographic Literature Review* 9.42 (1995), p. 784.

[12] Gustavo EAPA Batista, Xiaoyue Wang, and Eamonn J Keogh. “A complexity-invariant distance measure for time series”. In: *Proceedings of the 2011 SIAM international conference on data mining.* SIAM. 2011, pp. 699–710.

[13] David Baum et al. “Reading a phylogenetic tree: the meaning of monophyletic groups”. In: *Nature Education* 1.1 (2008), p. 190.

[14] Mark F Baumgartner and Sarah E Mussoline. “A generalized baleen whale call detection and classification system”. In: *The Journal of the Acoustical Society of America* 129.5 (2011), pp. 2889–2902.

[15] Mark F Baumgartner et al. “Glider-based passive acoustic monitoring in the Arctic”. In: *Marine Technology Society Journal* 48.5 (2014), pp. 40–51.

[16] Mark F Baumgartner et al. “Near real-time detection of low-frequency baleen whale calls from an autonomous surface vehicle: Implementation, evaluation, and remaining challenges”. In: *The Journal of the Acoustical Society of America* 149.5 (2021), pp. 2950–2962.

[17] Mark F Baumgartner et al. “Persistent near real-time passive acoustic monitoring for baleen whales from a moored buoy: System description and evaluation”. In: *Methods in Ecology and Evolution* 10.9 (2019), pp. 1476–1489.

[18] Mark F Baumgartner et al. "Real-time reporting of baleen whale passive acoustic detections from ocean gliders". In: *The Journal of the Acoustical Society of America* 134.3 (2013), pp. 1814–1823.

[19] Peter Beamish and Edward Mitchell. "Short pulse length audio frequency sounds recorded in the presence of a minke whale (Balaenoptera acutorostrata)". In: *Deep Sea Research and Oceanographic Abstracts*. Vol. 20. 4. Elsevier. 1973, pp. 375–386.

[20] Punam Bedi, Vinita Jindal, and Anjali Gautam. "Beginning with big data simplified". In: *2014 International Conference on Data Mining and Intelligent Computing (ICDMIC)*. IEEE. 2014, pp. 1–7.

[21] Justin D Bell, Reg A Watson, and Yimin Ye. "Global fishing capacity and fishing effort from 1950 to 2012". In: *Fish and Fisheries* 18.3 (2017), pp. 489–505.

[22] Jacob Benesty et al. *Noise reduction in speech processing*. Vol. 2. Springer Science & Business Media, 2009.

[23] Spencer K Lynn Bernd Würsig, Thomas A Jefferson, and Keith D Mullin. "Survey ships and aircraft". In: *Aquatic Mammals* 24.1 (1998), pp. 41–50.

[24] Lis Bittencourt et al. "Mapping cetacean sounds using a passive acoustic monitoring system towed by an autonomous Wave Glider in the Southwestern Atlantic Ocean". In: *Deep Sea Research Part I: Oceanographic Research Papers* 142 (2018), pp. 58–68.

[25] Ocean Studies Board, National Research Council, et al. *Marine mammal populations and ocean noise: determining when noise causes biologically significant effects*. National Academies Press, 2005.

[26] Steven Boll. "Suppression of acoustic noise in speech using spectral subtraction". In: *IEEE Transactions on acoustics, speech, and signal processing* 27.2 (1979), pp. 113–120.

[27] Transport Canada. *Interim order for the protection of North Atlantic right whales (eubalaena glacialis) in the Gulf of St. Lawrence, 2022.* Apr. 2022. URL: https://tc.canada.ca/en/ministerial-orders-interim-orders-directives-directions-response-letters/interim-order-protection-north-atlantic-right-whales-eubalaena-glacialis-gulf-st-lawrence-2022.

[28] Rich Caruana and Alexandru Niculescu-Mizil. "An empirical comparison of supervised learning algorithms". In: *Proceedings of the 23rd international conference on Machine learning.* 2006, pp. 161–168.

[29] Pierre Cauchy. "Ocean of sound: underwater gliders observing the oceanic environment". PhD thesis. University of East Anglia, 2021.

[30] Pierre Cauchy et al. "Passive Acoustic Monitoring from ocean gliders". In: *EGU General Assembly Conference Abstracts.* 2018, p. 430.

[31] Pierre Cauchy et al. "Sperm whale presence observed using passive acoustic monitoring from gliders of opportunity". In: *Endangered Species Research* 42 (2020), pp. 133–149.

[32] Pierre Cauchy et al. "Wind speed measured from underwater gliders using passive acoustics". In: *Journal of Atmospheric and Oceanic Technology* 35.12 (2018), pp. 2305–2321.

[33] Russell A Charif, Phillip J Clapham, and Christopher W Clark. "Acoustic detections of singing humpback whales in deep waters off the British Isles". In: *Marine Mammal Science* 17.4 (2001), pp. 751–768.

[34] Po-Hsuan Cameron Chen, Yun Liu, and Lily Peng. "How to develop machine learning models for healthcare". In: *Nature materials* 18.5 (2019), pp. 410–414.

[35] Yanping Chen et al. "Flying insect classification with inexpensive sensors". In: *Journal of insect behavior* 27.5 (2014), pp. 657–677.

[36] Yunjin Chen and Thomas Pock. "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration". In: *IEEE transactions on pattern analysis and machine intelligence* 39.6 (2016), pp. 1256–1272.

[37] Kyunghyun Cho et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation". In: *arXiv preprint arXiv:1406.1078* (2014).

[38] Francois Chollet. *Deep Learning with Python*. 1st. USA: Manning Publications Co., 2017. ISBN: 1617294438.

[39] Christopher W Clark. "Acoustic behavior of mysticete whales". In: *Sensory abilities of cetaceans*. Springer, 1990, pp. 571–583.

[40] Christopher W Clark, Moira W Brown, and Peter Corkeron. "Visual and acoustic surveys for North Atlantic right whales, Eubalaena glacialis, in Cape Cod Bay, Massachusetts, 2001–2005: Management implications". In: *Marine mammal science* 26.4 (2010), pp. 837–854.

[41] I. Cohen. "Optimal speech enhancement under signal presence uncertainty using log-spectral amplitude estimator". In: *IEEE Signal Processing Letters* 9.4 (2002), pp. 113–116.

[42] National Research Council. *Ocean Noise and Marine Mammals*. National Academies Press (US), 2003. ISBN: 978-0-309-08536-6.

[43] David R Cox. "The regression analysis of binary sequences". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 20.2 (1958), pp. 215–232.

[44] Gabriela Csurka. "Domain adaptation for visual applications: A comprehensive survey". In: *arXiv preprint arXiv:1702.05374* (2017).

[45] Robert Culkin and Sanjiv R Das. "Machine learning in finance: the case of deep learning for option pricing". In: *Journal of Investment Management* 15.4 (2017), pp. 92–100.

[46] Kostadin Dabov et al. "Image denoising by sparse 3-D transform-domain collaborative filtering". In: *IEEE Transactions on image processing* 16.8 (2007), pp. 2080–2095.

[47] Gretchen Daily et al. *Food production, population growth, and the environment.* 1998.

[48] Jaclyn N Daly and Jolie Harrison. "The Marine Mammal Protection Act: a regulatory approach to identifying and minimizing acoustic-related impacts on marine mammals". In: *The effects of noise on aquatic life.* Springer, 2012, pp. 537–539.

[49] Richard Davis et al. "Tracking whales on the Scotian Shelf using passive acoustic monitoring on ocean gliders". In: *OCEANS 2016 MTS/IEEE Monterey.* IEEE. 2016, pp. 1–4.

[50] Steven Davis and Paul Mermelstein. "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences". In: *IEEE transactions on acoustics, speech, and signal processing* 28.4 (1980), pp. 357–366.

[51] Houtao Deng et al. "A time series forest for classification and feature extraction". In: *Information Sciences* 239 (2013), pp. 142–153.

[52] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition.* Ieee. 2009, pp. 248–255.

[53] Weisheng Dong et al. "Nonlocally centralized sparse representation for image restoration". In: *IEEE transactions on Image Processing* 22.4 (2012), pp. 1620–1630.

[54] Yariv Ephraim and David Malah. "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator". In: *IEEE transactions on acoustics, speech, and signal processing* 33.2 (1985), pp. 443–445.

[55] Christine Erbe et al. "The Effects of Ship Noise on Marine Mammals—A Review". In: *Frontiers in Marine Science* 6 (2019), p. 606. ISSN: 2296-7745.

[56] Charles C Eriksen et al. "Seaglider: A long-range autonomous underwater vehicle for oceanographic research". In: *IEEE Journal of oceanic Engineering* 26.4 (2001), pp. 424–436.

[57] Alex C Ezeh, John Bongaarts, and Blessing Mberu. "Global population trends and policy options". In: *The Lancet* 380.9837 (2012), pp. 142–148.

[58] Linwei Fan et al. "Brief review of image denoising techniques". In: *Visual Computing for Industry, Biomedicine, and Art* 2.1 (2019), pp. 1–12.

[59] Hassan Ismail Fawaz et al. "Deep learning for time series classification: a review". In: *Data mining and knowledge discovery* 33.4 (2019), pp. 917–963.

[60] MC Ferguson et al. "Performance of manned and unmanned aerial surveys to collect visual data and imagery for estimating arctic cetacean density and associated uncertainty". In: *Journal of Unmanned Vehicle Systems* 6.3 (2018), pp. 128–154.

[61] NOAA Fisheries. *North Atlantic Right Whale.* URL: https://www.fisheries.noaa.gov/species/north-atlantic-right-whale.

[62] NOAA Fisheries. *North Atlantic Right Whale Conservation & Management.* URL: https://www.fisheries.noaa.gov/species/north-atlantic-right-whale#conservation-management.

[63] R Ewan Fordyce. "Cetacean evolution". In: *Encyclopedia of marine mammals.* Elsevier, 2018, pp. 180–185.

[64] Karin A Forney et al. "Nowhere to go: noise impact assessments for marine mammal populations with high site fidelity". In: *Endangered species research* 32 (2017), pp. 391–413.

[65] Hironobu Fujiyoshi, Tsubasa Hirakawa, and Takayoshi Yamashita. “Deep learning-based image recognition for autonomous driving”. In: *IATSS research* 43.4 (2019), pp. 244–252.

[66] Nikolas P Galatsanos, C Andrew Segall, and Aggelos K Katsaggelos. “Digital image enhancement”. In: *Encyclopedia of optical engineering* (2003), pp. 388–402.

[67] Timo Gerkmann and Richard C Hendriks. “Unbiased MMSE-based noise power estimation with low complexity and low tracking delay”. In: *IEEE Transactions on Audio, Speech, and Language Processing* 20.4 (2011), pp. 1383–1393.

[68] A. Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems.* O’Reilly Media, 2019. ISBN: 9781492032595. URL: https://books.google.co.uk/books?id=HnetDwAAQBAJ.

[69] Douglas Gillespie. “Detection and classification of right whale calls using an ‘edge’ detector operating on a smoothed spectrogram”. In: *Canadian Acoustics* 32.2 (2004), pp. 39–47.

[70] Douglas Gillespie et al. “Automatic detection and classification of odontocete whistles”. In: *The Journal of the Acoustical Society of America* 134.3 (2013), pp. 2427–2437.

[71] Douglas Gillespie et al. “PAMGUARD: Semiautomated, open source software for real-time acoustic detection and localisation of cetaceans”. In: *Journal of the Acoustical Society of America* 30.5 (2008), pp. 54–62.

[72] Xavier Glorot and Yoshua Bengio. “Understanding the difficulty of training deep feedforward neural networks”. In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics.* Ed. by Yee Whye Teh and Mike Titterington. Vol. 9. Proceedings of Machine Learning Research. Chia Laguna Resort, Sardinia, Italy: PMLR, 13–15 May 2010, pp. 249–256. URL: http://proceedings.mlr.press/v9/glorot10a.html.

[73] L. Gondara. “Medical Image Denoising Using Convolutional Denoising Autoencoders”. In: *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*. 2016, pp. 241–246. DOI: 10.1109/ICDMW.2016.0041.

[74] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Prentice Hall, 2002, pp. 91–94. ISBN: 9780201180756.

[75] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[76] Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. “Hybrid speech recognition with deep bidirectional LSTM”. In: *2013 IEEE workshop on automatic speech recognition and understanding*. IEEE. 2013, pp. 273–278.

[77] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. “Speech recognition with deep recurrent neural networks”. In: *2013 IEEE international conference on acoustics, speech and signal processing*. Ieee. 2013, pp. 6645–6649.

[78] Emily Greenhalgh. *Right Whale Consortium Releases 2020 Report Card Update*. URL: https://www.andersoncabotcenterforoceanlife.org/blog/2020-narwc-report-card/.

[79] Shuhang Gu et al. “Weighted nuclear norm minimization with application to image denoising”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 2862–2869.

[80] Philip Harding and Ben Milner. “Reconstruction-based speech enhancement from robust acoustic features”. In: *Speech Communication* 75 (2015), pp. 62–75.

[81] Simon Haykin. *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.

[82] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[83] Sindhu B Hegde et al. “Visual Speech Enhancement Without A Real Visual Stream”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2021, pp. 1926–1935.

[84] Hynek Hermansky. “Perceptual linear predictive (PLP) analysis of speech”. In: *the Journal of the Acoustical Society of America* 87.4 (1990), pp. 1738–1752.

[85] A. R. Hiby and P. S. Hammond. “Survey techniques for estimating abundance of cetaceans”. English. In: *Reports of the International Whaling Commission, Special Issue* 11 (Jan. 1989), pp. 47–80.

[86] Jon Hills et al. “Classification of time series by shapelet transformation”. In: *Data Mining and Knowledge Discovery* 28.4 (2014), pp. 851–881.

[87] Geoffrey Hinton et al. “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups”. In: *IEEE Signal Processing Magazine* 29.6 (2012), pp. 82–97.

[88] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. “A fast learning algorithm for deep belief nets”. In: *Neural computation* 18.7 (2006), pp. 1527–1554.

[89] Hans-Günter Hirsch and Christoph Ehrlicher. “Noise estimation techniques for robust speech recognition”. In: *1995 International conference on acoustics, speech, and signal processing*. Vol. 1. IEEE. 1995, pp. 153–156.

[90] S. Hochreiter and J. Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[91] Sepp Hochreiter. “The vanishing gradient problem during learning recurrent neural nets and problem solutions”. In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 6.02 (1998), pp. 107–116.

[92] Sepp Hochreiter et al. *Gradient flow in recurrent nets: the difficulty of learning long-term dependencies*. 2001.

[93] Kyaw Kyaw Htike and Othman O Khalifa. “Comparison of supervised and unsupervised learning classifiers for human posture recognition”. In: *International Conference on Computer and Communication Engineering (ICCCE’10)*. IEEE. 2010, pp. 1–6.

[94] Yi Hu and P.C. Loizou. “Subjective Comparison of Speech Enhancement Algorithms”. In: *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*. Vol. 1. 2006, pp. I–I. DOI: 10.1109/ICASSP.2006.1659980.

[95] Sergey Ioffe and Christian Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. In: *Proceedings of the 32nd International Conference on Machine Learning*. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, July 2015, pp. 448–456.

[96] S. Jarvis et al. “Automated Classification of Beaked Whales and Other Small Odontocetes in the Tongue of the Ocean, Bahamas”. In: *OCEANS 2006*. Sept. 2006, pp. 1–6. DOI: 10.1109/OCEANS.2006.307124.

[97] Susan M Jarvis et al. “Marine Mammal Monitoring on Navy Ranges (M3R): A toolset for automated detection, localization, and monitoring of marine mammals in open ocean environments”. In: *Marine Technology Society Journal* 48.1 (2014), pp. 5–20.

[98] Mark Johnson et al. “Beaked whales echolocate on prey”. In: *Proceedings of the Royal Society of London. Series B: Biological Sciences* 271.suppl_6 (2004), S383–S386.

[99] Biing Hwang Juang and Laurence R Rabiner. “Hidden Markov models for speech recognition”. In: *Technometrics* 33.3 (1991), pp. 251–272.

[100] Atakan Kantarci. *41 Statistics, Facts & Forecasts on Machine Learning [2021]*. Jan. 2021. URL: https://research.aimultiple.com/ml-stats/.

[101] Eamonn J Keogh and Michael J Pazzani. “Derivative dynamic time warping”. In: *Proceedings of the 2001 SIAM International Conference on Data Mining*. SIAM. 2001, pp. 1–11.

[102] Douglas A Kerr. “The ISO definition of the dynamic range of a digital still camera”. In: *see http://doug. kerr. home. att. net/pumpkin/ISO_Dynamic_range. pdf* (2008).

[103] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[104] Vijaya B Kolachalama and Priya S Garg. “Machine learning and medical education”. In: *NPJ digital medicine* 1.1 (2018), pp. 1–3.

[105] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems.* Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc., 2012, pp. 1097–1105.

[106] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems* 25 (2012), pp. 1097–1105.

[107] Madam Aravind Kumar and Kamsali Manjunatha Chari. “Noise Reduction Using Modified Wiener Filter in Digital Hearing Aid for Speech Signal Enhancement”. In: *Journal of Intelligent Systems* 29.1 (2020), pp. 1360–1378.

[108] Rajesh Kumar Rai, Puran Gour, and Balvant Singh. “Underwater image segmentation using clahe enhancement and thresholding”. In: *International Journal of Emerging Technology and Advanced Engineering* 2.1 (2012), pp. 118–123.

[109] Pedro Larrañaga et al. *Industrial applications of machine learning.* CRC press, 2018.

[110] Philip Leadbitter, Rob Hall, and Alexander Brearly. “A methodology for Thorpe scaling 512 Hz fast thermistor data from buoyancy-driven gliders to estimate turbulent kinetic energy dissipation rate in the ocean”. In: *OCEANS 2019 MTS/IEEE SEATTLE.* IEEE. 2019, pp. 1–5.

[111] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *nature* 521.7553 (2015), pp. 436–444.

[112] Stamatios Lefkimmiatis. “Non-local color image denoising with convolutional neural networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2017, pp. 3587–3596.

[113] Stephen E Levinson, Lawrence R Rabiner, and M Mohan Sondhi. “An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition”. In: *Bell System Technical Journal* 62.4 (1983), pp. 1035–1074.

[114] Tim Lewis et al. “Abundance estimates for sperm whales in the Mediterranean Sea from acoustic line-transect surveys”. In: (2018).

[115] Huang Lidong et al. “Combination of contrast limited adaptive histogram equalisation and discrete wavelet transform for image enhancement”. In: *IET Image Processing* 9.10 (2015), pp. 908–915.

[116] Jason Lines and Anthony Bagnall. “Time series classification with ensembles of elastic distance measures”. In: *Data Mining and Knowledge Discovery* 29.3 (2015), pp. 565–592.

[117] Jason Lines, Sarah Taylor, and Anthony Bagnall. “Time series classification with HIVE-COTE: The hierarchical vote collective of transformation-based ensembles”. In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 12.5 (2018), p. 52.

[118] Zhigang Ling et al. “Adaptive extended piecewise histogram equalisation for dark image enhancement”. In: *IET Image Processing* 9.11 (2015), pp. 1012–1019.

[119] Fan Liu, Qingzeng Song, and Guanghao Jin. “The classification and denoising of image noise based on deep neural networks”. In: *Applied Intelligence* (2020), pp. 1–14.

[120] Ming Liu et al. “Speech Enhancement Method Based On LSTM Neural Network for Speech Recognition”. In: *2018 14th IEEE International Conference on Signal Processing (ICSP)*. 2018, pp. 245–249. DOI: 10.1109/ICSP.2018.8652331.

[121] Xiaoxuan Liu et al. “A comparison of deep learning performance against healthcare professionals in detecting diseases from medical imaging: a systematic review and meta-analysis”. In: *The lancet digital health* 1.6 (2019), e271–e297.

[122] Philipos C Loizou. *Speech enhancement: theory and practice.* CRC press, 2007.

[123] Xugang Lu et al. “Speech enhancement based on deep denoising autoencoder.” In: *Interspeech.* Vol. 2013. 2013, pp. 436–440.

[124] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. “Rectifier nonlinearities improve neural network acoustic models”. In: *Proc. icml.* Vol. 30. 1. Citeseer. 2013, p. 3.

[125] Raman Maini and Himanshu Aggarwal. “A comprehensive review of image enhancement techniques”. In: *arXiv preprint arXiv:1003.4053* (2010).

[126] Julien Mairal et al. “Non-local sparse models for image restoration”. In: *2009 IEEE 12th international conference on computer vision.* IEEE. 2009, pp. 2272–2279.

[127] Robin A Makowski, D.R. McIntyre, and J.E. Heyning. *Cetacean Comparison Chart.* [Online; accessed April 27, 2017]. 2002. URL: https://www.acsonline.org.

[128] Philip’s Maps. *Philip’s World Atlas.* Philip’s World Atlas Series. Octopus Publishing Group, 2019. ISBN: 9781849075169.

[129] Pierre-François Marteau. “Time warp edit distance with stiffness adjustment for time series matching”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31.2 (2009), pp. 306–318.

[130] Rainer Martin. “Noise power spectral density estimation based on optimal smoothing and minimum statistics”. In: *IEEE Transactions on speech and audio processing* 9.5 (2001), pp. 504–512.

[131] Felix G Marx, Olivier Lambert, and Mark D Uhen. *Cetacean paleobiology*. John Wiley & Sons, 2016.

[132] Iacopo Masi et al. “Deep face recognition: A survey”. In: *2018 31st SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)*. IEEE. 2018, pp. 471–478.

[133] JN Matthews et al. “Vocalisation rates of the North Atlantic right whale (Eubalaena glacialis)”. In: *Journal of Cetacean Research and Management* 3.3 (2001), pp. 271–282.

[134] Megan F McKenna et al. “Morphology of the odontocete melon and its implications for acoustic function”. In: *Marine Mammal Science* 28.4 (2012), pp. 690–713.

[135] Tom McReynolds and David Blythe. *Advanced Graphics Programming Using OpenGL*. The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling. Morgan Kaufmann, 2005. ISBN: 9781558606593.

[136] David K Mellinger and Holger Klinck. *Passive autonomous acoustic monitoring of marine mammals with seagliders*. Tech. rep. Oregon State Univ Newport or Cooperative Inst for Marine Resources Studies, 2012.

[137] David K Mellinger, Aaron M Thode, and Anthony Martinez. “Passive acoustic monitoring of sperm whales in the Gulf of Mexico, with a model of acoustic detection distance”. In: *Proceedings of the twenty-first annual Gulf of Mexico information transfer meeting*. 2002, pp. 493–501.

[138] David K Mellinger et al. “Passive acoustic monitoring in the Northern Gulf of Mexico using ocean gliders”. In: *The Journal of the Acoustical Society of America* 142.4 (2017), pp. 2533–2533.

[139] David K. Mellinger. “A comparison of methods for detecting right whale calls”. en-US. In: *Canadian Acoustics* 32.2 (June 2004), pp. 55–65. ISSN: 2291-1391. (Visited on 02/22/2019).

[140] David K. Mellinger and Christopher W. Clark. “Recognizing transient low-frequency whale sounds by spectrogram correlation”. In: *The Journal of the Acoustical Society of America* 107.6 (May 2000), pp. 3518–3529. ISSN: 0001-4966. (Visited on 02/22/2019).

[141] Stefan Meyer et al. “Marine mammal population decline linked to obscured by-catch”. In: *Proceedings of the National Academy of Sciences* 114.44 (2017), pp. 11781–11786.

[142] D. Michelsanti et al. “Deep-learning-based audio-visual speech enhancement in presence of Lombard effect”. In: *Speech Communication* 115 (2019), pp. 38–50. ISSN: 0167-6393. DOI: https://doi.org/10.1016/j.specom.2019.10.006.

[143] B. Milner. “A comparison of front-end configurations for robust speech recognition”. In: *ICASSP*. 2002, pp. 797–800.

[144] Ben Milner. “Enhancing speech at very low signal-to-noise ratios using non-acoustic reference signals”. In: *Speech Communication* 55.9 (2013), pp. 879–892.

[145] Marvin Minsky and Seymour A Papert. *Perceptrons: An introduction to computational geometry*. MIT press, 2017.

[146] Tom M. Mitchell. *Machine Learning*. New York: McGraw-Hill, 1997. ISBN: 978-0-07-042807-2.

[147] Michael Moore and Mark Baumgartner. *Saving The North Atlantic Right Whale*. Woods Hole Oceanographic Institution (WHOI), 2020. DOI: 10.1575/1912/24708.

[148] Xavier Mouy, Mohammed Bahoura, and Yvan Simard. “Automatic recognition of fin and blue whale calls for real-time monitoring in the St. Lawrence”. In: *The Journal of the Acoustical Society of America* 126 (Dec. 2009), pp. 2918–28. DOI: 10.1121/1.3257588.

[149] Vinod Nair and Geoffrey E Hinton. “Rectified linear units improve restricted boltzmann machines”. In: *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*. 2010, pp. 807–814.

[150] Vinod Nair and Geoffrey E Hinton. “Rectified linear units improve restricted boltzmann machines”. In: *Icml*. 2010.

[151] Tiago Nazaré et al. *Deep Convolutional Neural Networks and Noisy Images*. Springer, Jan. 2018, pp. 416–424.

[152] Chigozie Nwankpa et al. “Activation functions: Comparison of trends in practice and research for deep learning”. In: *arXiv preprint arXiv:1811.03378* (2018).

[153] Emmanuel Okewu, Philip Adewole, and Oladipupo Sennaike. “Experimental comparison of stochastic optimizers in deep learning”. In: *International Conference on Computational Science and Its Applications*. Springer. 2019, pp. 704–715.

[154] Hui Ou et al. “Automated extraction and classification of time-frequency contours in humpback vocalizations”. In: *The Journal of the Acoustical Society of America* 133.1 (2013), pp. 301–310.

[155] Ashutosh Pandey and DeLiang Wang. “A new framework for CNN-based speech enhancement in the time domain”. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 27.7 (2019), pp. 1179–1188.

[156] Jim R Parker. *Algorithms for image processing and computer vision*. John Wiley & Sons, 2010.

[157] Susan E Parks et al. “Characteristics of gunshot sound displays by North Atlantic right whales in the Bay of Fundy”. In: *The Journal of the Acoustical Society of America* 131.4 (2012), pp. 3173–3179.

[158] Susan E Parks et al. “Individual right whales call louder in increased environmental noise”. In: *Biology letters* 7.1 (2011), pp. 33–35.

[159] Vanessa Pirotta et al. “Consequences of global shipping traffic for marine giants”. In: *Frontiers in Ecology and the Environment* 17.1 (2019), pp. 39–47.

[160] Stephen M Pizer et al. “Adaptive histogram equalization for automatic contrast enhancement of medical images”. In: *Application of Optical Instrumentation in Medicine XIV and Picture Archiving and Communication Systems*. Vol. 626. International Society for Optics and Photonics. 1986, pp. 242–250.

[161] Stephen M. Pizer et al. “Adaptive histogram equalization and its variations”. In: *Computer Vision, Graphics, and Image Processing* 39.3 (1987), pp. 355–368. ISSN: 0734-189X. DOI: https://doi.org/10.1016/S0734-189X(87)80186-X. URL: https://www.sciencedirect.com/science/article/pii/S0734189X8780186X.

[162] Daniel Povey et al. “The Kaldi speech recognition toolkit”. In: *IEEE 2011 workshop on automatic speech recognition and understanding*. CONF. IEEE Signal Processing Society. 2011.

[163] Aneesur Rahman and Frank H Stillinger. “Propagation of sound in water. A molecular-dynamics study”. In: *Physical Review A* 10.1 (1974), p. 368.

[164] Bharath Ramsundar and Reza Bosagh Zadeh. *TensorFlow for deep learning: from linear regression to reinforcement learning*. " O’Reilly Media, Inc.", 2018.

[165] Sundarrajan Rangachari and Philipos C Loizou. “A noise-estimation algorithm for highly non-stationary environments”. In: *Speech communication* 48.2 (2006), pp. 220–231.

[166] Tariq Rashid. *Make your own neural network*. CreateSpace Independent Publishing Platform, 2016.

[167] Aaron N. Rice et al. “Variation of ocean acoustic environments along the western North Atlantic coast: A case study in context of the right whale migration route”. In: *Ecological Informatics* 21 (2014). Ecological Acoustics, pp. 89–99. ISSN: 1574-9541. DOI: https://doi.org/10.1016/j.ecoinf.2014.01.005.

[168] Marie A Roch et al. “Gaussian mixture model classification of odontocetes in the Southern California Bight and the Gulf of California”. In: *The Journal of the Acoustical Society of America* 121.3 (2007), pp. 1737–1748.

[169] Marie A. Roch et al. “Comparison of machine learning techniques for the classification of echolocation clicks from three species of odontocetes”. en-US. In: *Canadian Acoustics* 36.1 (Mar. 2008), pp. 41–47. ISSN: 2291-1391. URL: https://jcaa.caa-aca.ca/index.php/jcaa/article/view/1989 (visited on 02/22/2019).

[170] Fanny Roche et al. “Autoencoders for music sound modeling: a comparison of linear, shallow, deep, recurrent and variational models”. In: (2019).

[171] Callum Rollo et al. “Glider observations of the Northwestern Iberian margin during an exceptional summer upwelling season”. In: *Journal of Geophysical Research: Oceans* 125.8 (2020), e2019JC015804.

[172] Karen Rose, Scott Eldridge, and Lyman Chapin. “The internet of things: An overview”. In: *The internet society (ISOC)* 80 (2015), pp. 1–50.

[173] F. Rosenblatt. *The Perceptron, a Perceiving and Recognizing Automaton Project Para.* Report: Cornell Aeronautical Laboratory. Cornell Aeronautical Laboratory, 1957.

[174] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. *Learning internal representations by error propagation.* Tech. rep. California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[175] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. “Learning representations by back-propagating errors”. In: *nature* 323.6088 (1986), pp. 533–536.

[176] Olga Russakovsky et al. “ImageNet Large Scale Visual Recognition Challenge”. In: *International Journal of Computer Vision (IJCV)* 115.3 (2015), pp. 211–252. DOI: 10.1007/s11263-015-0816-y.

[177] A. Samuel. “Some Studies in Machine Learning Using the Game of Checkers”. In: *IBM J. Res. Dev.* 3 (1959), pp. 210–229.

[178] J. M. Sanches, J. C. Nascimento, and J. S. Marques. “Medical Image Noise Reduction Using the Sylvester–Lyapunov Equation”. In: *IEEE Transactions on Image Processing* 17.9 (2008), pp. 1522–1539. DOI: 10.1109/TIP.2008.2001398.

[179] Björn Schuller et al. “Recognition of noisy speech: A comparative survey of robust model architecture and feature enhancement”. In: *EURASIP Journal on Audio, Speech, and Music Processing* 2009 (2009), pp. 1–17.

[180] Sequoia Scientific. *LISST-Glider*. URL: https://www.sequoiasci.com/product/lisst-glider/.

[181] M. L. Seltzer, D. Yu, and Y. Wang. “An investigation of deep neural networks for noise robust speech recognition”. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. 2013, pp. 7398–7402.

[182] Samsu Sempena, Nur Ulfa Maulidevi, and Peb Ruswono Aryan. “Human action recognition using dynamic time warping”. In: *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics*. IEEE. 2011, pp. 1–5.

[183] Pierre Sermanet et al. “Overfeat: Integrated recognition, localization and detection using convolutional networks”. In: *arXiv preprint arXiv:1312.6229* (2013).

[184] Agung W Setiawan et al. “Color retinal image enhancement using CLAHE”. In: *International Conference on ICT for Smart Society*. IEEE. 2013, pp. 1–3.

[185] Charles Severance. “Eben upton: Raspberry pi”. In: *Computer* 46.10 (2013), pp. 14–16.

[186] Alex Sherstinsky. “Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network”. In: *Physica D: Nonlinear Phenomena* 404 (2020), p. 132306.

[187] Shaohuai Shi et al. “Benchmarking state-of-the-art deep learning software tools”. In: *2016 7th International Conference on Cloud Computing and Big Data (CCBD)*. IEEE. 2016, pp. 99–104.

[188] Yu Shiu et al. “Deep neural networks for automated detection of marine mammal species”. In: *Scientific reports* 10.1 (2020), pp. 1–12.

[189] Karen Simonyan and Andrew Zisserman. “Two-stream convolutional networks for action recognition in videos”. In: *arXiv preprint arXiv:1406.2199* (2014).

[190] Karen Simonyan and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *International Conference on Learning Representations*. 2015.

[191] Evgeny Smirnov. “North atlantic right whale call detection with convolutional neural networks”. In: *Proc. Int. Conf. on Machine Learning, Atlanta, USA*. Citeseer. 2013, pp. 78–79.

[192] Melissa S Soldevilla et al. “Passive acoustic monitoring on the North Atlantic right whale calving grounds”. In: *Endangered Species Research* 25.2 (2014), pp. 115–140.

[193] University of St Andrews. *DCLDE 2013. Scottish Oceans Institute.* URL: https://soi.st-andrews.ac.uk/dclde2013/.

[194] United States. *The Endangered Species Act as amended by Public Law 97-304 (the Endangered Species Act amendments of 1982).* 1983.

[195] Alexandra Stefan, Vassilis Athitsos, and Gautam Das. “The move-split-merge metric for time series”. In: *IEEE Transactions on Knowledge and Data Engineering* 25.6 (2013), pp. 1425–1438.

[196] Baochen Sun, Jiashi Feng, and Kate Saenko. “Return of frustratingly easy domain adaptation”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 30. 1. 2016.

[197] Jalal Taghia et al. “An evaluation of noise power spectral density estimation algorithms in adverse acoustic environments”. In: *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2011, pp. 4640–4643.

[198] Ke Tan and DeLiang Wang. “A Convolutional Recurrent Neural Network for Real-Time Speech Enhancement.” In: *Interspeech*. 2018, pp. 3229–3233.

[199] Adi L Tarca et al. “Machine learning and its applications to biology”. In: *PLoS computational biology* 3.6 (2007), e116.

[200] The Naval Technology. *C-Enduro Autonomous Surface Vehicle*. URL: https://www.naval-technology.com/projects/c-enduro-autonomous-surface-vehicle/.

[201] Rini Smita Thakur, Ram Narayan Yadav, and Lalita Gupta. “State-of-art analysis of image denoising methods using convolutional neural networks”. In: *IET Image Processing* 13.13 (2019), pp. 2367–2380.

[202] Chunwei Tian, Yong Xu, and Wangmeng Zuo. “Image denoising using deep CNN with batch renormalization”. In: *Neural Networks* 121 (2020), pp. 461–473.

[203] George Toderici et al. “Variable rate image compression with recurrent neural networks”. In: *arXiv preprint arXiv:1511.06085* (2015).

[204] Eric Tzeng et al. “Adversarial discriminative domain adaptation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 7167–7176.

[205] Cornell University. *Cornell Bioacoutics NRW Monitoring*. URL: https://portal.nrwbuoys.org/ab/dash/.

[206] Cornell University. *The Marinexplore and Cornell University Whale Detection Challenge*. URL: https://www.kaggle.com/c/whale-detection-challenge/data.

[207] Ildar R Urazghildiiev and Christopher W Clark. “Acoustic detection of North Atlantic right whale contact calls using the generalized likelihood ratio test”. In: *The Journal of the Acoustical Society of America* 120.4 (2006), pp. 1956–1963.

[208] Ildar R Urazghildiiev et al. “Detection and recognition of North Atlantic right whale contact calls in the presence of ambient noise”. In: *IEEE Journal of Oceanic Engineering* 34.3 (2009), pp. 358–368.

[209] Régis Vaillant, Christophe Monrocq, and Yann Le Cun. “Original approach for the localisation of objects in images”. In: *IEE Proceedings-Vision, Image and Signal Processing* 141.4 (1994), pp. 245–250.

[210] A. Varga and H.J.M. Steeneken. “Assessment for automatic speech recognition II: NOISEX-92: a database and an experiment to study the effect of additive noise on speech recognition systems”. In: *Speech Communication* 12.3 (1993), pp. 247–251.

[211] Ursula K Verfuss et al. “A review of unmanned vehicles for the detection and monitoring of marine fauna”. In: *Marine Pollution Bulletin* 140 (2019), pp. 17–29.

[212] W Vickers, B Milner, and R Lee. “Improving The Robustness Of Right Whale Detection In Noisy Conditions Using Denoising Autoencoders And Augmented Training”. In: *ICASSP 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2021, pp. 91–95.

[213] W. Vickers et al. “Detecting right whales from autonomous surface vehicles using RNNs and CNNs”. In: *2019 27th European Signal Processing Conference Workshop (EUSIPCO)*. IEEE. 2019.

[214] William Vickers et al. “A comparison of machine learning methods for detecting right whales from autonomous surface vehicles”. In: *2019 27th European Signal Processing Conference (EUSIPCO)*. IEEE. 2019, pp. 1–5.

[215] William Vickers et al. “Methods to Improve the Robustness of Right Whale Detection using CNNs in Changing Conditions”. In: *2020 28th European Signal Processing Conference (EUSIPCO)*. IEEE. 2021, pp. 106–110.

[216] William Vickers et al. “Robust North Atlantic right whale detection using deep learning models for denoising”. In: *The Journal of the Acoustical Society of America* 149.6 (2021), pp. 3797–3812.

[217] Pascal Vincent et al. “Extracting and composing robust features with denoising autoencoders”. In: *Proceedings of the 25th international conference on Machine learning*. 2008, pp. 1096–1103.

[218] Pascal Vincent et al. “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion.” In: *Journal of machine learning research* 11.12 (2010).

[219] Auriane Virgili et al. “Combining multiple visual surveys to model the habitat of deep-diving cetaceans at the basin scale: Large-scale modelling of deep-diving cetacean habitats”. In: *Global Ecology and Biogeography* 28.3 (2019), pp. 300–314.

[220] Athanasios Voulodimos et al. “Deep learning for computer vision: A brief review”. In: *Computational intelligence and neuroscience* 2018 (2018).

[221] Li Wan et al. “Regularization of neural networks using dropconnect”. In: *International conference on machine learning*. PMLR. 2013, pp. 1058–1066.

[222] Mei Wang and Weihong Deng. “Deep visual domain adaptation: A survey”. In: *Neurocomputing* 312 (2018), pp. 135–153.

[223] Lindy S Weilgart. “The impacts of anthropogenic ocean noise on cetaceans and implications for management”. In: *Canadian journal of zoology* 85.11 (2007), pp. 1091–1116.

[224] Tsung-Hsien Wen et al. “Semantically conditioned lstm-based natural language generation for spoken dialogue systems”. In: *arXiv preprint arXiv:1508.01745* (2015).

[225] Christopher Whitt et al. “Future vision for autonomous ocean observations”. In: *Frontiers in Marine Science* 7 (2020), p. 697.

[226] Thorsten Wuest et al. “Machine learning in manufacturing: advantages, challenges, and applications”. In: *Production & Manufacturing Research* 4.1 (2016), pp. 23–45.

[227] Bernd Würsig, William F Perrin, and JGM Thewissen. *Encyclopedia of marine mammals*. Academic Press, 2009.

[228] Bernd Würsig, William F Perrin, and JGM Thewissen. *Encyclopedia of marine mammals*. Academic Press, 2009, pp. 1173–1179.

[229] Zimmer Walter M X. *Passive acoustic monitoring of Cetaceans*. Cambridge University Press, Apr. 2011, pp. 115–116. DOI: 10.1017/CBO9780511977107.

[230] Zimmer Walter M X. *Passive acoustic monitoring of Cetaceans*. Cambridge University Press, Apr. 2011, pp. 1–2. DOI: 10.1017/CBO9780511977107.

[231] Zimmer Walter M X. *Passive acoustic monitoring of Cetaceans*. Cambridge University Press, Apr. 2011, pp. 1–356. DOI: 10.1017/CBO9780511977107.

[232] Jianbin Xiong et al. “Application of Histogram Equalization for Image Enhancement in Corrosion Areas”. In: *Shock and Vibration* 2021 (2021).

[233] Wayne Xiong et al. “Achieving human parity in conversational speech recognition”. In: *arXiv preprint arXiv:1610.05256* (2016).

[234] Jun Xu et al. “Stacked Sparse Autoencoder (SSAE) for Nuclei Detection on Breast Cancer Histopathology Images”. In: *IEEE Transactions on Medical Imaging* 35.1 (2016), pp. 119–130. DOI: 10.1109/TMI.2015.2458702.

[235] Tina M Yack et al. “Comparison of beaked whale detection algorithms”. In: *Applied Acoustics* 71.11 (2010), pp. 1043–1049.

[236] Tina M Yack et al. “Passive acoustic monitoring using a towed hydrophone array results in identification of a previously unknown beaked whale habitat”. In: *The Journal of the Acoustical Society of America* 134.3 (2013), pp. 2589–2595.

[237] Matthew D Zeiler and Rob Fergus. “Visualizing and understanding convolutional networks”. In: *European conference on computer vision*. Springer. 2014, pp. 818–833.

[238] Kai Zhang, Wangmeng Zuo, and Lei Zhang. “FFDNet: Toward a fast and flexible solution for CNN-based image denoising”. In: *IEEE Transactions on Image Processing* 27.9 (2018), pp. 4608–4622.

[239] Kai Zhang et al. “Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising”. In: *IEEE Transactions on Image Processing* 26.7 (2017), pp. 3142–3155. DOI: 10.1109/TIP.2017.2662206.

[240] Kai Zhang et al. “Learning Deep CNN Denoiser Prior for Image Restoration”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017.

[241] Xiaofan Zhang et al. “Fusing Heterogeneous Features From Stacked Sparse Autoencoder for Histopathological Image Analysis”. In: *IEEE Journal of Biomedical and Health Informatics* 20.5 (2016), pp. 1377–1383. DOI: 10.1109/JBHI.2015.2461671.

[242] Youshan Zhang. “A Survey of Unsupervised Domain Adaptation for Visual Recognition”. In: *arXiv preprint arXiv:2112.06745* (2021).

[243] Jianchao Zhou et al. “Robust sound event classification by using denoising autoencoder”. In: *2016 IEEE 18th International Workshop on Multimedia Signal Processing (MMSP)*. 2016, pp. 1–6. DOI: 10.1109/MMSP.2016.7813376.

[244] Ali Ziaei et al. “A novel approach for contrast enhancement in biomedical images based on histogram equalization”. In: *2008 International Conference on BioMedical Engineering and Informatics*. Vol. 1. IEEE. 2008, pp. 855–858.

[245] Karel Zuiderveld. *Contrast Limited Adaptive Histogram Equalization*. USA: Academic Press Professional, Inc., 1994, pp. 474–485. ISBN: 0123361559.

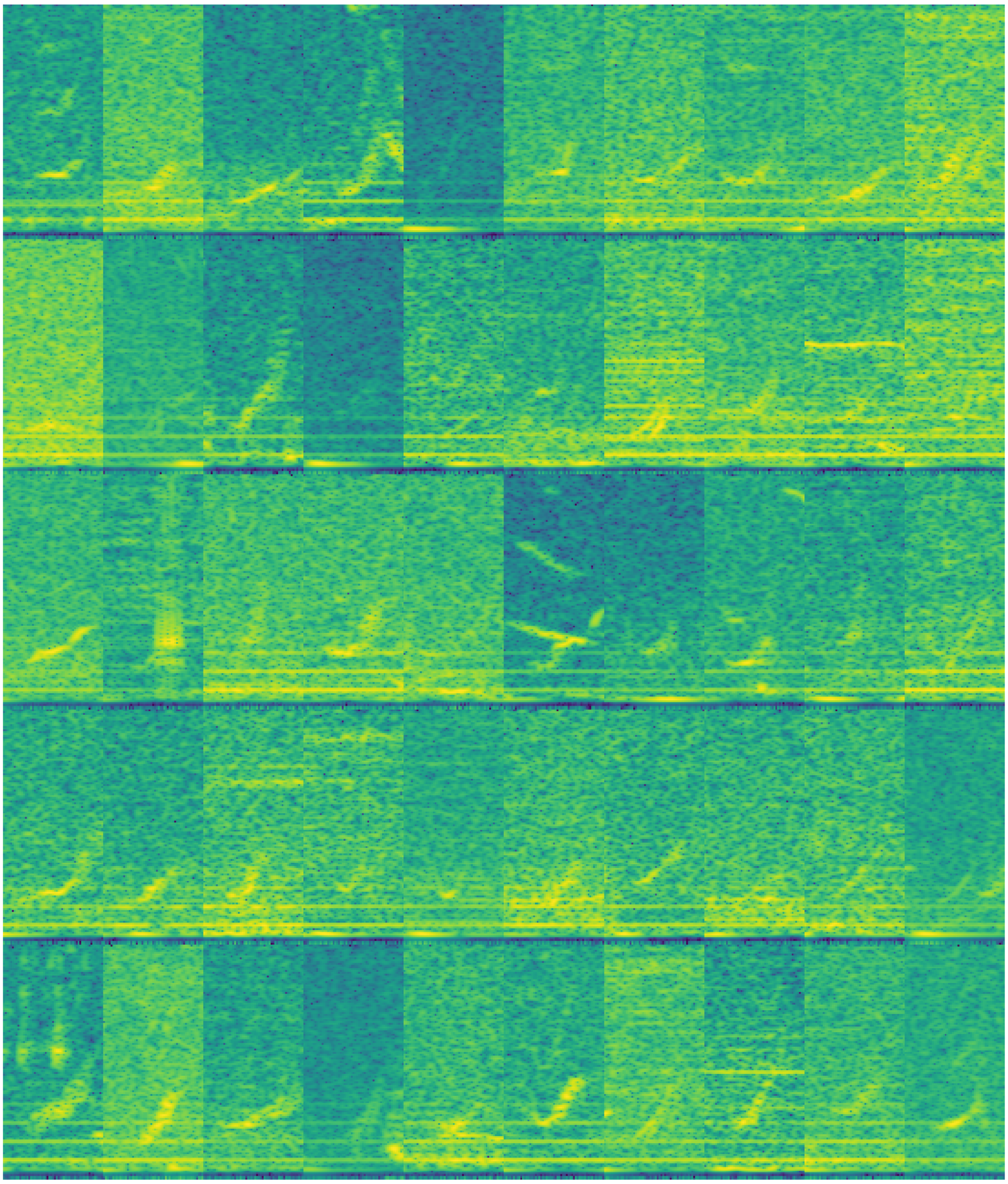
Appendix A

A.1

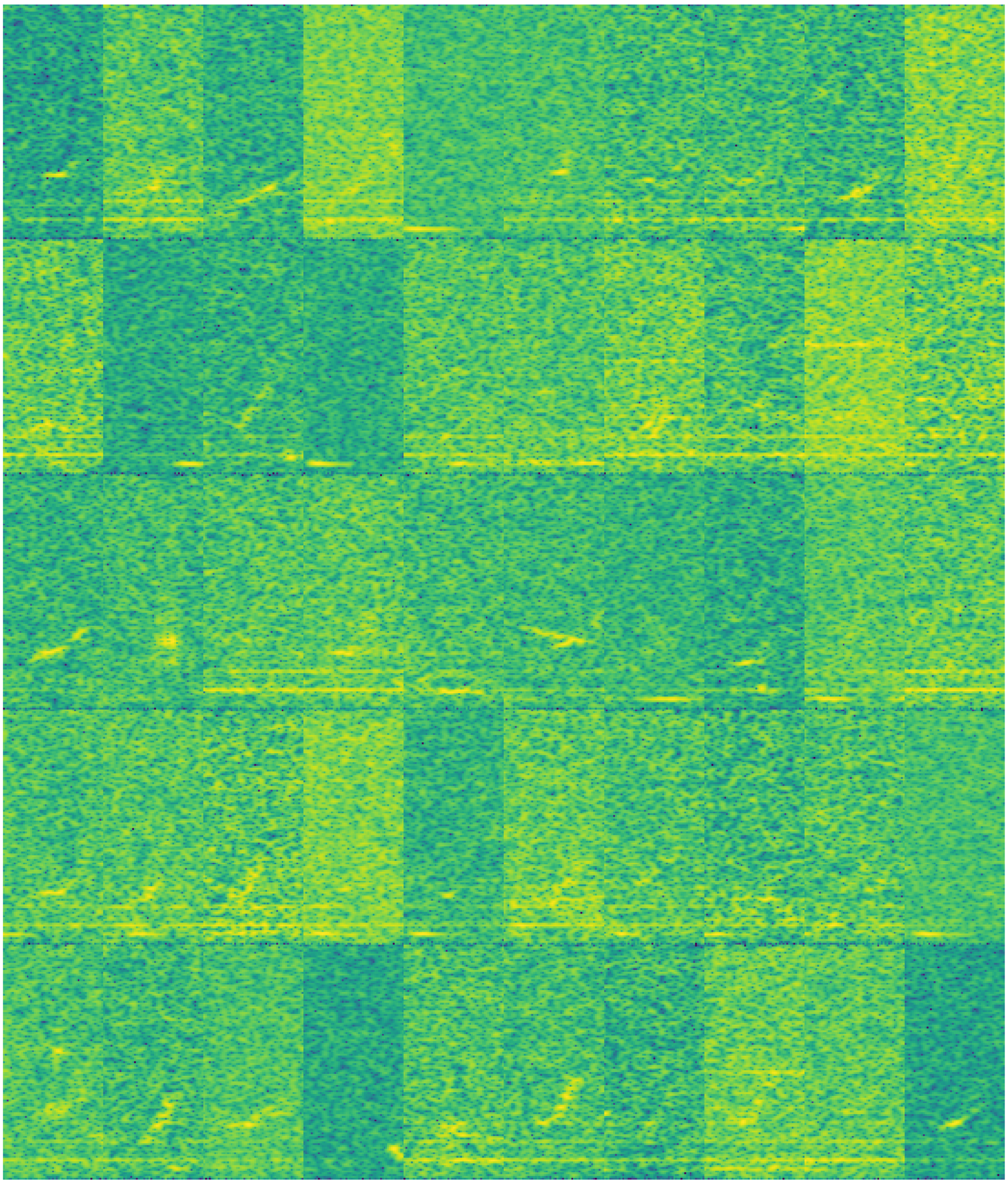
		SNR (dB)			
		5	0	-5	-10
Encoder depth (layers)	1	89.74%	82.14%	68.86%	57.19%
	2	**90.35%**	84.04%	75.26%	61.14%
	3	89.37%	**85.07%**	**75.39%**	62.56%
	4	89.41%	84.05%	75.05%	**64.15%**
	5	87.49%	82.57%	71.88%	61.89%
	6	87.08%	81.36%	72.42%	62.36%

Table A.1 An extended table of all DAE test results. All tests were run with the addition of white noise at displayed SNRs levels. All DAE architectures were fixed other than the encoder (and subsequently, decoder) depths, which are shown.

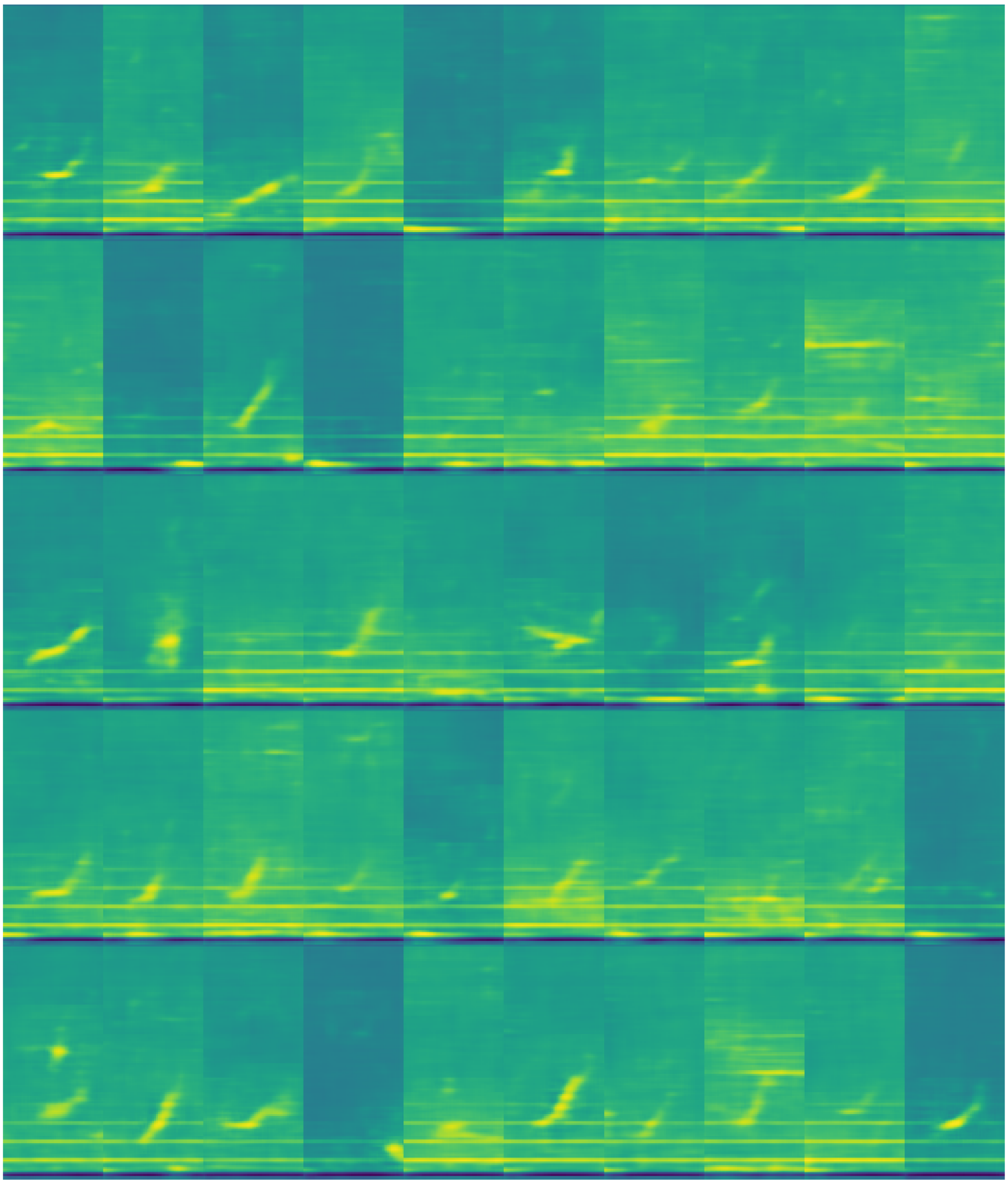
A.2



(a) Original vocalisations from Stellwagen corpus



(b) Matched vocalisations to (a), corrupted with white noise at -5dB.



(c) Spectrograms from (b) denoised via the DAE.

Fig. A.1 50 spectrograms from the test set of Stellwagen. Figure (a) shows the original spectrograms. Figure (b) shows the same spectrograms with the addition of white noise at -5dB. Figure (c) shows the denoising spectrograms after processing via the DAE detailed in Chapter 5.3.2.