MASTER THESIS
Preventing shipping traffic collisions through identification and classification of the North Atlantic right whale bioacoustic signatures

Project Report

Mirjana Erceg

Aalborg University Copenhagen
Architecture, Design, and Media Technology

Architecture, Design, and Media Technology
Aalborg University
http://www.aau.dk

AALBORG UNIVERSITY
STUDENT REPORT

Title:
MASTER THESIS
Preventing shipping traffic collisions through identification and classification of the North Atlantic right whale bioacoustic signatures

Theme:
Signal processing and Deep learning

Project Period:
Spring Semester 2022

Project Group:

Participant(s):
Mirjana Erceg

Supervisor(s):
George Palamas

Copies: 1

Page Numbers: 79

Date of Completion:
June 8, 2022

Abstract:

Whales play an essential role in sustaining the ecological balance of the world. Despite this, the North Atlantic right whale population is declining rapidly due to vessel collisions, commercial whaling, bycatch, entanglement, climate change and debris, with only about 400 left. Researchers have achieved very good results in identifying and classifying sounds using Data augmentation and Deep learning. Research presented here discusses the use of NARW's sounds to avoid vessel collisions by identifying nearby NARWs and classifying their bioacoustic signatures. As part of this project, sounds are converted into spectrograms that are fed into a Convolutional Neural Network. The data was augmented with time warping, frequency masking, and time masking, and Transfer learning was used for training. It has been found that the dataset was effectively trained with an accuracy of 88%. This model may ultimately lead to even better results in the future if improvements are made.

Architecture, Design, and Media Technology
Aalborg Universitet
http://www.aau.dk

AALBORG UNIVERSITET
STUDENTERRAPPORT

Titel:
MASTER THESIS
Preventing shipping traffic collisions through identification and classification of the North Atlantic right whale bioacoustic signatures

Tema:
Signal processing and Deep learning

Projektperiode:
Forårssemester 2022

Projektgruppe:

Deltager(e):
Mirjana Erceg

Vejleder(e):
George Palamas

Oplagstal: 1

Sidetal: 79

Afleveringsdato:
8. juni 2022

Abstract:

Her er resuméet

Contents

Chapter 1

Introduction

The environmental concerns surrounding the largest (ocean) animals on the planet are a permanent item on the agenda of many conservation groups. In order to thrive, all animals require a secure and healthy environment. The International Whaling Commission (IWC) points out that rapid habitat degradation is a major cause of species declines. Humans directly affect the animal habitat by making it unsuitable for animals, such as cetaceans and other marine life.[1]

These concerns are not new, as they have existed for hundreds of years or longer, probably since the 1800s. Scientist Sir William Henry Flower, an English surgeon, zoologist, and director of the Natural History Museum in London who was well known for his expertise on mammals, is thought to have been the first one to articulate clear concerns about the sustainability of whaling.[2] A quote from him can be found in one of his works, "For countless centuries impulses from within and the forces of circumstances from without have been gradually shaping the whales into their present wonderful form and gigantic size, but the very perfection of their structure and their magnitude combined, the rich supply of oil protecting their internal parts from cold, the beautiful apparatus of whalebone by which their nutrition is provided for, have been fatal gifts, which, under the sudden revolution produced on the surface of the globe by the development of the wants and arts of civilised man, cannot but lead in a few years to their extinction. It does not need much foresight to divine the future history of whales...".[3]

According to the "Recent memoirs on the Cetacea"[4], whaling by many nations during the 16th to 19th centuries was described as a cruel, inhumane killing practice that almost led to extermination of whales in the seas of Spitzbergen and other regions.

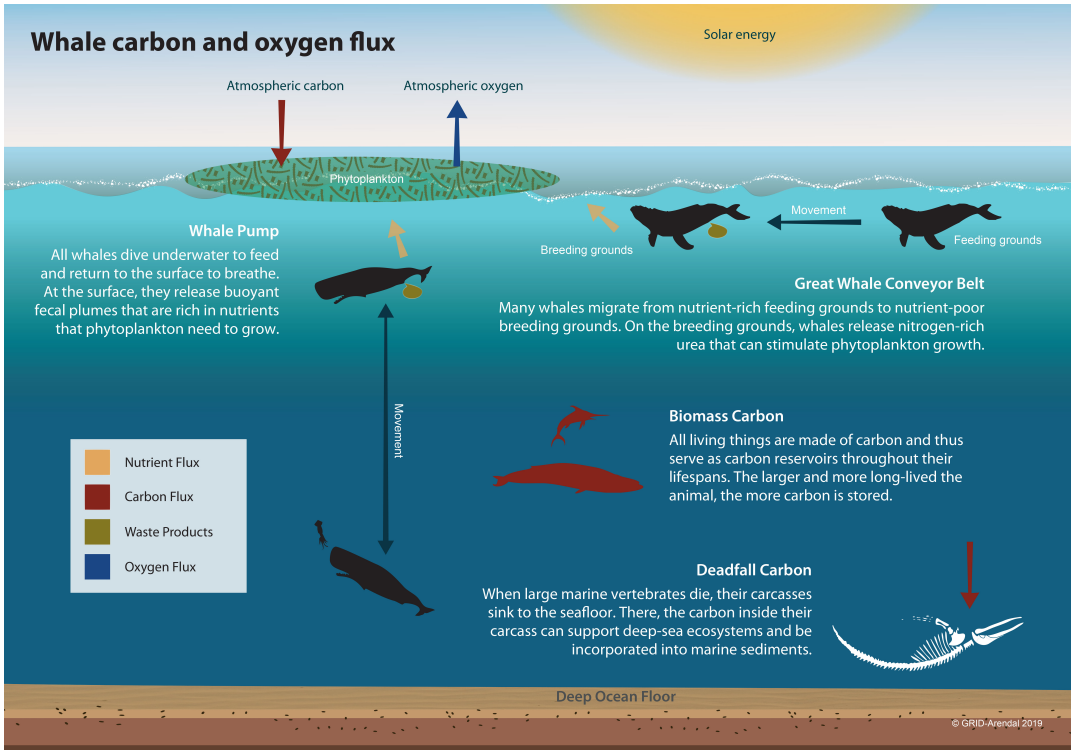
Besides serving as a carbon sink and producing food for billions of people, the ocean is the largest ecosystem on Earth. It has taken marine science a long time to acknowledge the growing body of evidence that apex predators play a crucial role in managing natural ecosystems. There is evidence to suggest that long-lived large

species could maintain ecosystem balance. Whales contribute a range of services to the ecosystem, making them valuable. Through direct predation, they utilize significant thropic control on the marine environment, causing physical changes and affecting species biodiversity. In ocean ecosystems, the oscillations caused by disorder in climate, primary productivity and predation are stabilised by whales. Removing the whales would likely lead to a significant disturbance in the marine ecosystem.[5][6].

Phytoplankton blooms as the primary production are enhanced through abundant discharge of iron and nitrogen from whale's urine and faeces. Phytoplankton is hugely important as it has a main function in the production of approximately 50% of all oxygen while at the same time it captures nearly 40% of all carbon dioxide (CO_2) produced. Having whales contributes to a wider distribution of oxygen and nutrients which in turn causes a growth of primary production rich in prey abundance and biodiversity.[5][6][7]

Figure 1.1 displays whale carbon and oxygen flux.

Figure 1.1: Whale carbon and oxygen flux [7]



Several studies report that whales have a positive impact on primary production, nutrients transfer to the surface, and CO_2 capture. Upon death, their carcasses sink to the bottom of the ocean floor, delivering a large quantity of nutrients to the nutrient-scarce parts of the ocean while storing thousands of tons of carbon.[6]

In the aftermath of centuries of commercial whaling, the majority of whale species are critically endangered, endangered, threatened, depleted or protected.[8] Climate change, specifically ocean acidification that has been linked to food chain disruption, entanglement in fishing gear, bycatch, oil and gas activities, shipping, noise, habitat degradation, chemical pollution, and marine debris are additional stressors that threaten whales.[9][10]

Whales and oceans play important roles in the environment as well as an important economic engine that offers services vital to a prosperous economy, which in turn sustains life on Earth. The ocean, particularly its life-giving ecosystem, is generally overlooked and regarded as having little formal value, as Delloite explains in its report.[11] We tend to focus on the removal of monetary value from the ocean, but we have been oblivious to its continued devastation and the considerable higher value that is accessible through sustainable ocean economy management. The authors also warn if we continue to only extract resources from the ocean without considering the environmental consequences, it will all inevitably result in ecological and economic collapse. According to best estimates, under 1% of the ocean's overall value has been committed to sustainable ocean projects, making the UN's Sustainable Development Goal (SDG) - Life below water the least funded SDG of all.

Gathering data at great depths and under high pressures is a logistical and scientific challenge because of the shifting and uncompromising nature of the open ocean and deep sea. In fact, the deep ocean is the least explored environment on Earth, with 90% of the species collected there being new to science. It is challenging to conserve cetaceans because there is a great deal of information we don't know about them. According to a Canadian oceanographer Dr Paul Snelgrove, "we know more about the surface of the Moon and about Mars than we do about this habitat". In order to solve the above problems, national governments, industries, organizations, local communities, and conservation groups need to work together. Some of the most successful campaigns take years to develop.[10][11]

Because whales are enormous and their environment is vast, tracking them individually would be impossible. Monitoring techniques have been employed to observe human impacts on marine ecosystems, such as passive acoustic monitoring (PAM) and aerial or ship-based image data collection. There are many factors that determine the quality of visual studies performed, including daylight, weather conditions, and the availability of suitable research platforms. Continuous monitoring of large areas and difficult-to-reach areas is possible with PAM. With a large number of PAM recorders installed at various sites, a mass-scale surveillance network can be developed to monitor the acoustic habitats of a multitude of species over time.[12]

Underwater, sound propagates more effectively than light, so marine species have evolved to communicate, navigate, and scout prey using acoustic signals. The

North Atlantic right whale (Eubalaena glacialis, NARW) is one of the species that PAM closely monitors.[13] There are less than 400 estimated NARWs living in the North Atlantic waters, making it one of the most critically endangered whales in the world and their numbers are rapidly declining.[14][15][16] Over the last 10 years, the population has declined by 26 percent and the calving rate has dropped by nearly 40 percent.[17][18]

Following are chapters dedicated to describing the findings of eco-biodiversity research, state of the art technology and its implementation, obtained results, and recommendations for future work. A more detailed discussion on cetaceans, in particular NARWs, is found in Chapter 2. After that, the environmental stressors causing their rapid declines and possible conservation solutions are presented, followed by a review of the state of the art to support the technologies chosen for this project. The theories behind the technologies are discussed in Chapter 3: audio signal processing and Transfer learning with a pretrained Neural Network. A description is presented in Chapter 4 concerning the Machine Learning tools and libraries used, the dataset and the procedures used to process it, Data augmentation techniques, and the structure of the Machine Learning model. Chapter 5 presents and discusses the results. The conclusion in Chapter 6 describes personal motivation, discusses the accomplishment of learning goals, and sets goals for the future, followed by a discussion in Chapter 7 on how to improve future research efforts. Chapter 8 is dedicated to acknowledging those who contributed to the creation of this project. The Appendix A - Environmental stressors includes other environmental stressors, the Appendix B - Code includes the code, and the Appendix C - Training results concludes with the rest of the findings.

1.1 Problem statement

“To protect the North Atlantic right whale species from colliding with shipping traffic, as well as to raise awareness of how population declines are affecting the species.”

As part of this project, which aims to bring attention to NARW population declines, research, analysis, and methods are used to improve detection and recognition of NARW calls using Transfer Learning and Google’s pretrained model "GoogleLeNet". The objectives of this study are to determine whether the model’s training accuracy will be satisfactory or highly promising after the research has been completed and to see what applications can be created using the results.

Chapter 2

Analysis

2.1 Ecology and biodiversity

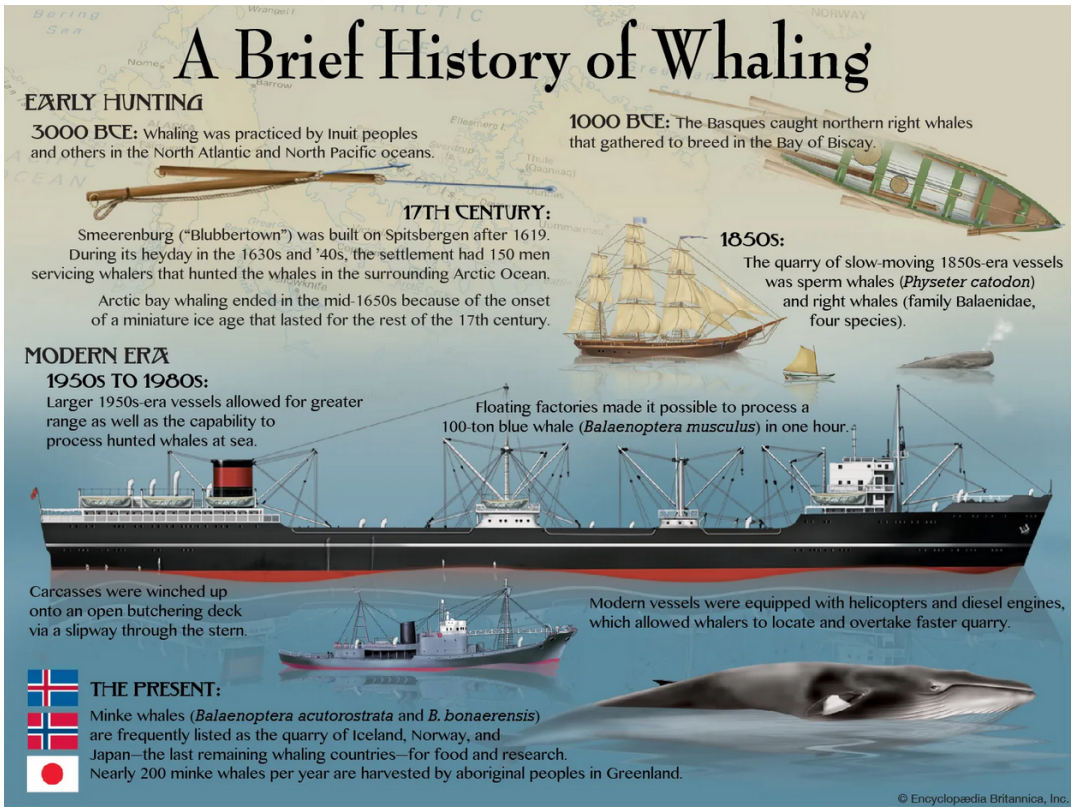
Whales have evolved from land ancestors some 40 million years ago to become fully aquatic. The first whaling evidence dates back to 6000BCE in South Korea. Several places where whaling took place have been identified, including Greenland, Norway and islands off the coast of Japan. They hunted whales for food, tools, and fuel. Whales were hunted by the Basques (Spain) in the 11th century for their oil and baleen, also known as "whalebone". More than a hundred thousand whales were killed in the United States in the mid-19th century for their valuable oil. Sperm whales produced spermaceti (special liquid found in their heads) and ambergris (a waxy substance that was used for perfume and medicine).[19] Figure 2.1 shows a brief history of whaling.

There were many things made from baleen: carriage springs, clothes, fishing equipment, whips and other. According to estimates, between 1900 and 1999, the whaling industry managed to kill tens of millions of whales due to modernized vessels and other advances in machinery and technology.[6][19]

It is vital to point out that up until the early 20th century, scarcely more than baleen and blubber was used, and most of the carcasses were thrown away. The mid-20th century witnessed whale populations record a dramatic decline, which was accompanied by a reduction in cruel whaling practices.[20] There were 2.9 million whales removed, one of the largest removals of any animal biomass ever recorded on the planet. An overview of the removals of different whale species in the Northern and Southern Hemispheres can be found in Figure 2.2. As a result of World War II, whaling declined in the Southern Hemisphere during that time, which was actually beneficial for whales.[21][22]

NARWs have been commercially exploited for nearly 1.000 years, and although whaling has mostly ceased, there are still many threats posed by humans. The eastern population of this species is considered extinct, but historically, there was

Figure 2.1: A brief history of whaling [20]

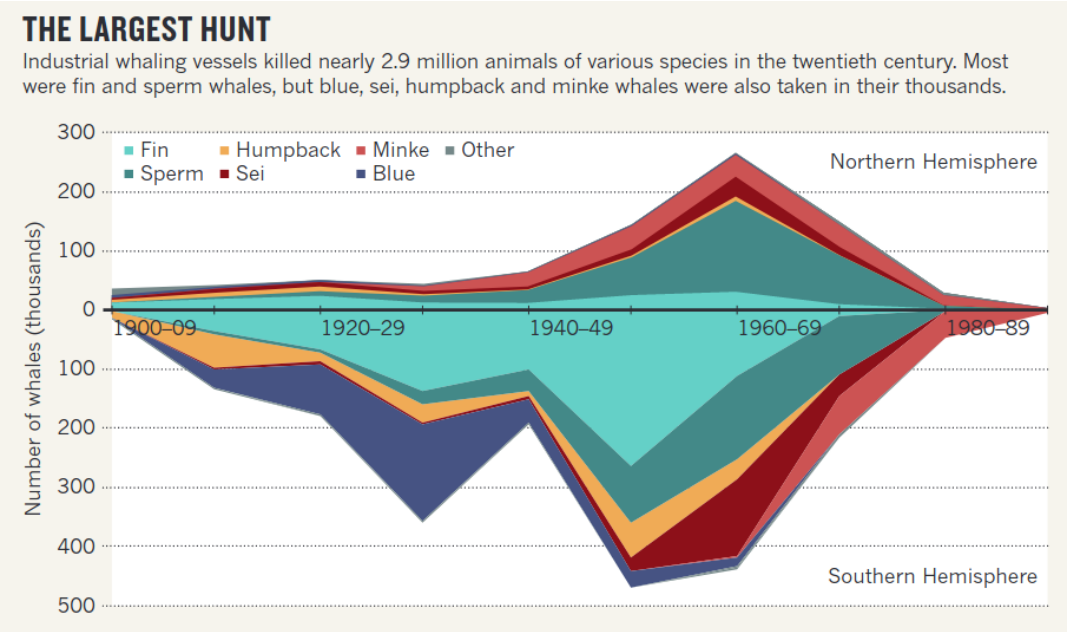


a population on each side of the North Atlantic. The early catches in the western North Atlantic have not been quantified, but it has been estimated that between 1634-1950, over 5.500-11.000 NARWs were killed. Although whale numbers gradually increased after the ban on whaling, recent studies indicate that they have declined again. [18]

Whales are generally defined as cetaceans longer than three meters. A total of 90 species of whales, dolphins and porpoises are recognized at present. Dolphins and porpoises are also members of the Cetacea order. Two suborders of cetaceans comprise it: Mysticeti (baleen whales or mysticetes - approximately 14 species) and Odontceti (toothed whales or odontocetes - approximately 76 species).[18]

Whales can grow up to 200 tons in weight and reach lengths of 30 meters (blue whale), making them the largest and heaviest animals on earth. They are found in all oceans and seas of the world. Despite having flippers and tail flukes similar to fish, whales are mammals. Due to the fact that they are covered in blubber, which acts as an insulation layer to protect from hypothermia, the large whales generate excessive heat and therefore have thermoregulating systems to prevent overheating.[23]

Figure 2.2: The largest hunt [22]

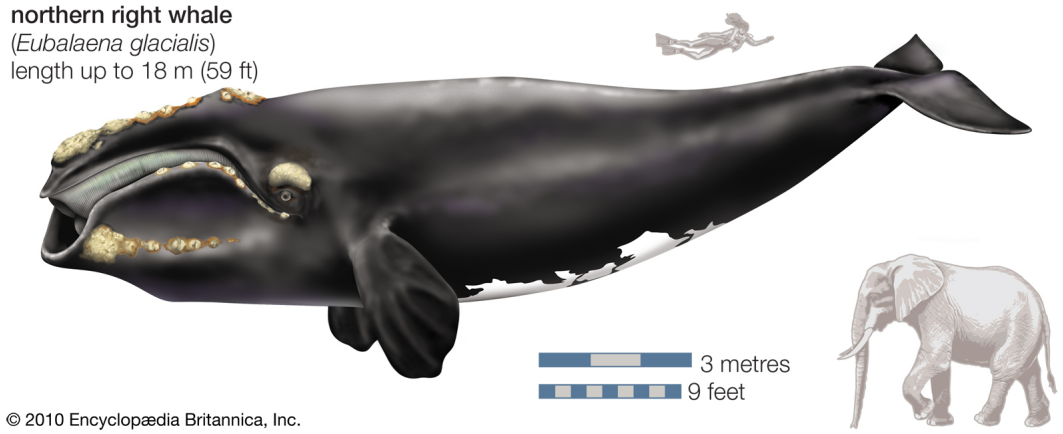


Monitoring whales can be difficult, so little is known about their real lifespan, although whales often live for at least 20 years. The lifespan of a Bowhead whale ranges from 20 to 100 years, with some living up to 200 years.[19]

Figure 2.3 - NARWs are baleen whales and can grow up to 18 meters tall and weigh up to 90 tons, with females typically measuring 1 meter longer than males. NARWs could live up to 70 years, up to 85 in prime conditions, and around 35 years today due to man-made stressors. The increased stress causes females to give birth only once around every 10.2 years. Scientists could distinguish these whales by their varied patterns of white patches of rough skin on their heads (callosities), which help differentiate them from each other. The tails on these whales are long and their underside has highly variable white patches.[18] Aerial photos taken over the past 20 years were analyzed by Stewart et al. [24] for changes in NARWs body length. As a result of the study, the maximum length of the 2019 calf will be shorter by about a meter when compared to the 1981 calf, representing a decline of 7.3 percent. Body changes might indicate a population collapse caused by environmental stressors.

Whales can recognize one another and belong from small to medium sized social groups. During breeding and feeding, several pods gather together. A variety of signals are used including pulsed songs, clicks, groans, songs, whistles, low rumbles, and body language. Whales rely on sound to communicate as sound travels four times faster through water than through air. Many social groups use their own distinctive sound to communicate. Baleen whales are renowned for produc-

Figure 2.3: North Atlantic right whale [25]



ing complex songs that are considered to be one of the most sophisticated forms of communication among mammals. During the breeding season, these songs are possibly used to communicate with offspring, to entice mates, to mark territories, and even to coordinate migrations. During the mating season, some whales sing the same songs, but their patterns change from year to year.[19][26]

Toothed whales use sound and echolocation to explore their environment and communicate under water since they have limited vision. Rosenkvist et al. [27] conducted a study to see if humans are able to use echolocation similarly to certain mammals. In the study, users navigated the virtual environment using visualized echolocation signals. Sound signal visualization helped the participants of the study to navigate the unlit virtual environment. It also enabled them to mentally map out the virtual world. Though it may be true that toothed whales and humans have similar surface-level characteristics when it comes to sound production and recognition, the toothed whale's echolocation is superior, therefore, their physiological mechanisms for sounds differ considerably from those of humans.

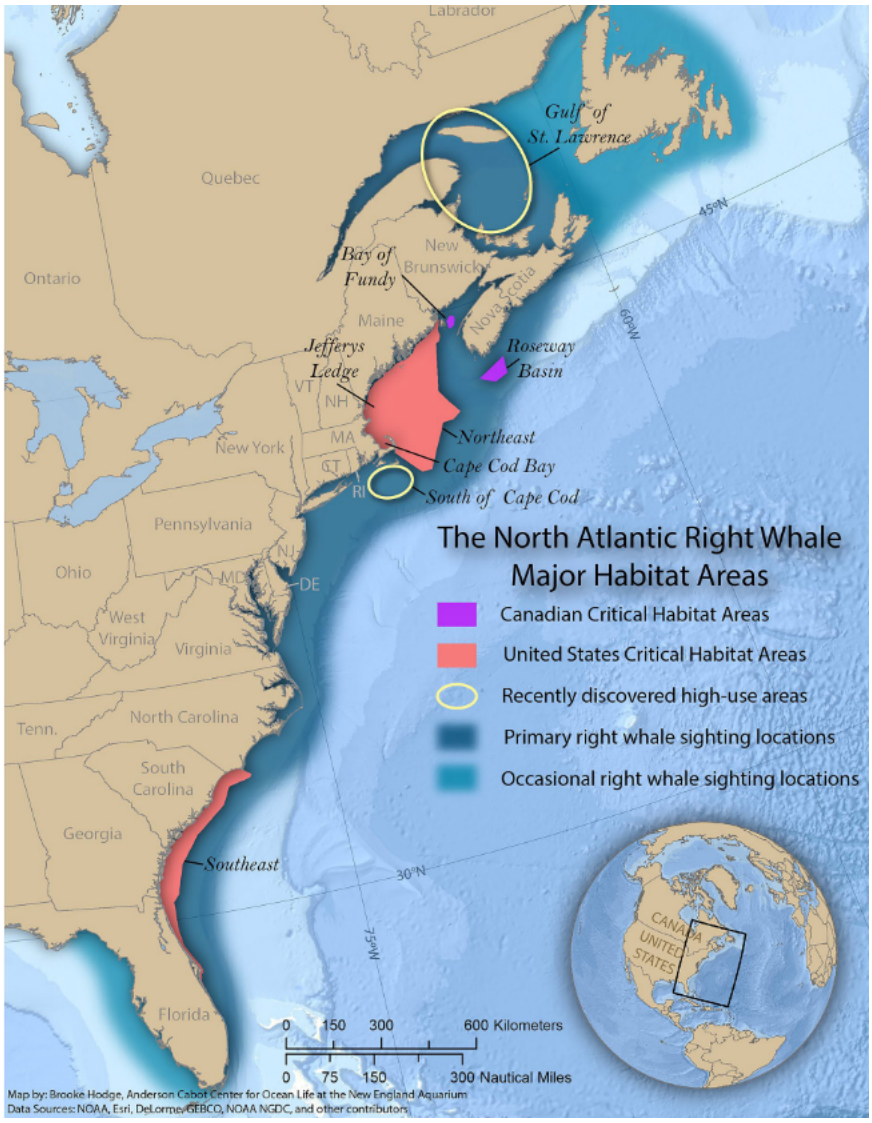
Whales have evolved hearing to recognize underwater sound vibrations. Some evidence suggests that the 10Hz sounds of fin whales can travel over 1.800 km. Baleen whales are experts at hearing low-frequency sounds used for long-distance communication.[23][19]

In addition to their highly vocal nature, NARWs produce a wide variety of low-frequency sounds. These 1-2 second "upcalls," which are believed to be used for communication over long distances, are the most common sounds. The variations in the upcalls can be used to identify individuals, according to recent research. [18]

NARWs generally moves slowly and may spend a considerable amount of time resting at the surface. In most cases, they are active at the surface, and they do not show much fear of boats, but instead may be curious and approachable when near. Additionally, they sleep on the surface. From the Gulf of St. Lawrence in the north

to Florida in the south, there is only one remaining population of these species in the western North Atlantic.[10][15][16][18] Figure 2.4 illustrates the critical habitats and other high-use areas of the North Atlantic right whale.

Figure 2.4: NARWs major habitat areas [15]



Aside from the fact that their population is growing very slowly, they face two major threats today - vessel strikes and entanglement in fishing gear, this will be discussed in detail in the next section. Additionally, there are other threats such as noise pollution, commercial whaling, bycatch, climate change, and marine debris.

2.1.1 Environmental stressors

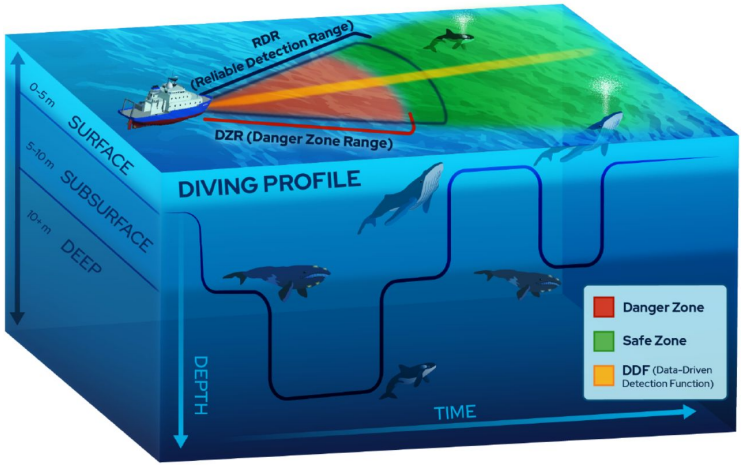
Vessel strikes and noise pollution

Due to their habitat near the coast, NARWs congregate around busy ports and have a tendency to move slowly. This makes them particularly vulnerable to vessel strikes. Due to their tendency to feed near the surface, and the lack of a dorsal fin, they are difficult to spot, and worse yet, they tend to gather in the shipping lanes. According to National Oceanic and Atmospheric Administration's (NOAA) report, 34 NARWs have been killed during the period 2017-2021 due to ship strikes or entanglement.[19][18][28]

Figure 2.5 illustrates the relevant areas and whale dive states. There is a 20-degree field of view in a surface-based detection system which divides a detection area into a danger zone and a safe zone. These zones are determined by the vessel speed, reaction time, and detection range of the mitigation system. The whales dive in three layers based on their depth: surface [0-5m], subsurface [5-10m] and deep [10m+]. A whale can only be identified by a vessel if it is near the surface and blowing.[29]

Normally, NARWs do feeding dives that last between 10 and 20 minutes, and sometimes up to 40 minutes. Most of their dives are shallow and near the surface, but some dives may reach depths of 200 meters or more.[18]

Figure 2.5: Important areas and whale dive states [29]

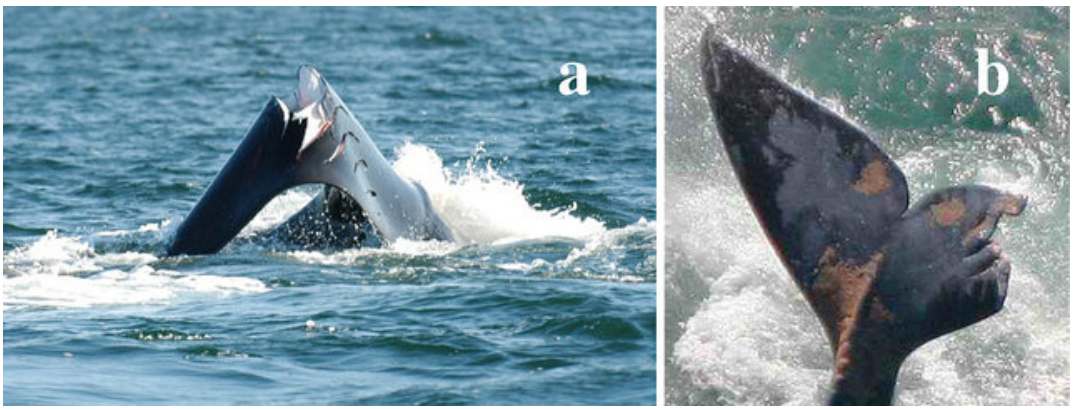


Typically, collisions or strikes occur when marine animals come into contact with parts of a boat (most often the propeller or bow), resulting in blunt force trauma or even death of the animal. If the vessel is significantly damaged in a collision, there can be serious injuries or death for those onboard (vessel crew). Animals other than whales are also affected by ship strikes. Data indicates that 75 species have been killed by ship strikes, according to Schoeman et al..[30]

Injury results in a poor quality of life for animals due to stress, pain, and potentially negative psychological conditions. Despite not fully understanding the long-term effects of strikes, a number of species display reduced fitness and locomotive impairments. A causal relationship can exist between the high mortality rate and the low rate of population growth.[30] The wounds on NARWs in Figure 2.6 and Figure 2.7 are both caused by propellers.[31]

Figure 2.6: Observation of large propeller wounds and resolving scars along the dorsolateral aspect of the torso of a North Atlantic right whale [31]

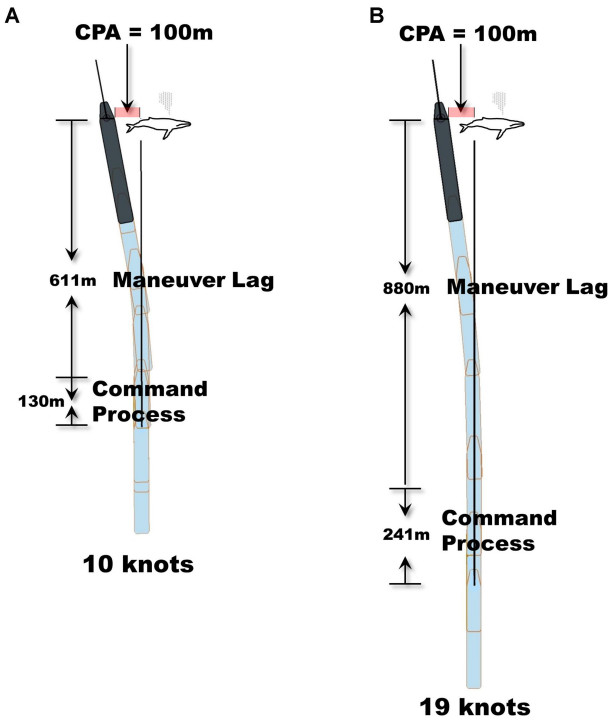
Figure 2.7: An incised NARW fluke: (a) March 10, 2005, (b) September 3, 2005 [31]



The International Whaling Commission (IWC) has identified 14 high-risk areas: places where a high number of vessels and animals overlap. There is a great deal of focus on big vessels, since they pose a greater risk to whales. When vessels travel at a high rate of speed, the impact is stronger and serious injuries are more likely. To illustrate, reducing the speed of large vessels to 10 knots lowered the possibility of a disastrous impact fatality to less than 50%. Even if the vessel operator is able to see the animals in time, maneuvering and determining the distance to the animals are crucial for impact avoidance.[30]

In their research, Gende et al. [32] analysed active whale avoidance carried out by large vessels, a procedure that is typically initiated by the mariner when they spot a whale surface. Researchers concluded that active avoidance is possible and can be conducted without increasing of risks. Figure 2.8 illustrates the manoeuvring capabilities of a large cruise ship traveling at 10 and 19 knots when approaching within 100 meters. In ideal circumstances, a boat traveling at 10 knots would have to turn 741 meters from the whale. However, one traveling at 19 knots would have to do so at least 1121 meters away.

Figure 2.8: The maneuvering capabilities of a large cruise ship travelling at 10 knots (A) and 19 knots (B) [32]



Why the animals do not flee from approaching vessels remains one of the most challenging questions to answer. While foraging, socialising, nursing, and resting, the animals are likely distracted from risk detection, and it is possible that they

cannot hear incoming vessels.[30] Because whales are notoriously unresponsive to incoming ships, any action that enables whale avoidance should be developed.[32] Over half of the 40 NARWs necropsied between 1970 and 2006 died from ship strikes, which may be responsible for up to ten NARW deaths each year.[14] Many collisions with large vessels go undetected and unreported.[33] It is possible for the shipping industry to oppose speed restrictions due to the economic implications of the extra time at sea caused by lower speeds. This is applicable to large areas and it may contribute to the low compliance with voluntary speed reductions.[32]

An individual from from A.P. Møller – Mærsk A/S, the largest transport and logistics company in the world, was interviewed in order to gain a deeper understanding of the problem with shipping traffic. In his role as the Head of Marine Standards, Mr Aslak Ross was able to provide some insights.

There was a question about the policies and procedures the company follows when on the ocean. Mr Ross replied that the company strictly adheres to the regulations based on the International Convention for the Prevention of Pollution from Ships (MARPOL) adopted by the International Maritime Organisation (IMO). The designed policies and procedures ensure compliance with MARPOL by the shipping industry. Furthermore, he explained that there are very strict rules regarding the filtering and disposal of discharges. In relation to the zero waste policy, the only thing allowed is food waste that is shredded onboard. To address specific measures, he explained that NOAA, which identifies all protected areas and restricted zones and assists in the enforcement of them, and Mærsk are working together to strengthen their procedures. They are trying to make it as easy as possible for their ships, since the information they receive is very local. Whenever a ship fails to follow certain guidelines, the ruling government bodies issue a warning. As the biggest shipping company in the world, they took the initiative to work with the World Wildlife Fund (WWF), where they discussed what were their thoughts and concerns about biodiversity conservation. Similarly, Mærsk launched a scoping project with the WWF, some universities, and other NGOs in order to better understand how their vessels produce noise. Could solutions be found to reduce it, and are they performing measurements on their own or including naval forces? As part of the company's oceans and landslides activities in marine protected areas in 2022, the company hopes to build technologies to enable better monitoring of ships and to keep a check on assessments of the newly built terminals.

The Solution possibilities for conservation section discusses in detail the policies and regulations proposed by the IMO and IWC.

People are making the ocean noisier as a result of their activities. In the marine environment, it poses a growing concern due to its negative effects on marine mammals, invertebrates, and fish. Breeze et al. [34], for example, indicate that Canadian marine waters are home to many noise-producing activities, such as seismic surveying (and other oil and gas activities), recreational boating, pile

driving, and military activities. Shipping traffic in some of North America's busiest ports creates significant noise pollution because shipping traffic overlaps with endangered ocean species.

Researchers Stanistreet et al. [35] have reported that beaked and bottlenose whales in several regions around the world responded strongly to active naval sonar signals by avoiding them. Consequently, these sounds are perceived as potential threats by beaked whales. Affected diving patterns and moving away from the source were observed during controlled experimental exposures, as well as an end to foraging. Even while diving to great depths, the whales stopped producing echolocation clicks, switching to non-foraging behavior. The use of active sonar in naval exercises has been associated with decompression sickness in stranded animals. While sonar signals were being used, whales' acoustic activity reduced (e.g. sperm whale clicks), indicating that foraging was halted, sometimes for several hours. There was a shift in orientation, direction, dive profiles, and acoustic behavior as animals moved away from the exposure site.

As reported by Matthews and Parks [36], North Atlantic right whales display the following acoustic behavior. To compensate for the effects of a decrease in communication space, individual callers must shift their signals, such as frequency, duration, or amplitude. Studies conducted on right whales in North Atlantic and Southern habitats indicate that individuals produce higher-frequency calls at higher levels of noise. There was a decrease in NARW stress hormones in Bay of Fundy during the low-noise period following the September 11th attack, demonstrating that noise could stress NARWs. Although this is the case, researchers cannot determine whether hormone levels and noise levels were linked.

Please refer to the Appendix A - Environmental stressors for a more detailed explanation on other environmental stressors.

2.1.2 Possible conservation solutions

As a specialized agency of the United Nations, the IMO is responsible for maritime safety and security and the prevention of maritime and atmospheric pollution emitted by vessels. The IMO regulates routing and reporting measures or speed restrictions.[37][38]

IMO states the only effective mitigation measure is avoiding areas where whales are known to congregate and reducing speed when passing those special areas. In a number of studies, the IWC has found that high speed is associated with an increased risk, which supports speed restrictions as a risk reduction method. A mandatory speed reduction of 10 knots was enacted in the NARW habitat, which greatly reduced the number of strikes.[39]

In order to prevent strikes, the IWC recognizes that decreasing geographical overlaps between cetaceans and vessels remains the most effective solution. To

reduce collision probabilities, they strive to use collision-reducing measures on a global scale, such as reducing/limiting vessel speed and redirecting shipping lanes. Moreover, they plan to improve reporting incidents, expand collaboration on vessel strike problems internationally, develop preventive/avoidance technologies, and raise industry and public awareness. The IWC and IMO manage the vessel strikes database on a global scale. This is to better understand the extent of the problem, provide specific advice to the shipping industry, as well as maintain efforts for better, more accurate reporting of strike incidents. There are various educational materials and software applications available too, such as Global Fishing Watch, WhaleSENSE, Whale Alert, and Automatic Identification System, and others. It is crucial to continue to identify high risk areas and whale species that are small or declining.[33]

In addition to routing to alternative routes without compromising navigation safety, coastal states can propose route changes that are outside their territorial waters in order to avoid ship strikes. Educating and training the crew is another way to prevent collisions. A propeller guard could be installed around a propeller as a physical boundary, but further research is needed. So far, none of the deterrent devices tested have been effective in keeping the whales away from the vessels.[30]

The activities of humans in their primary habitats and along their migration routes must be limited in order to minimise human-induced noise pollution. Furthermore, reducing ocean noise pollution will contribute to the survival of this majestic species by making whale populations bounce back more easily.[26]

Whaling is governed by the IWC for both commercial and indigenous purposes, but not all countries are members or follow the corresponding rules. Native whalers are required to report their catch to the IWC. By bringing pressure to these governments involved in these cruel practices, NGOs and the public may succeed in banning commercial whaling for all time.[40]

The only way to solve the problem of by-catch and entanglement is to collaborate with small and large fishing communities, focusing on the environmental impact of by-catch and entanglement. As it can be dangerous to try to release or rescue large animals, first responders should be well trained to handle distressed and injured animals.[40]

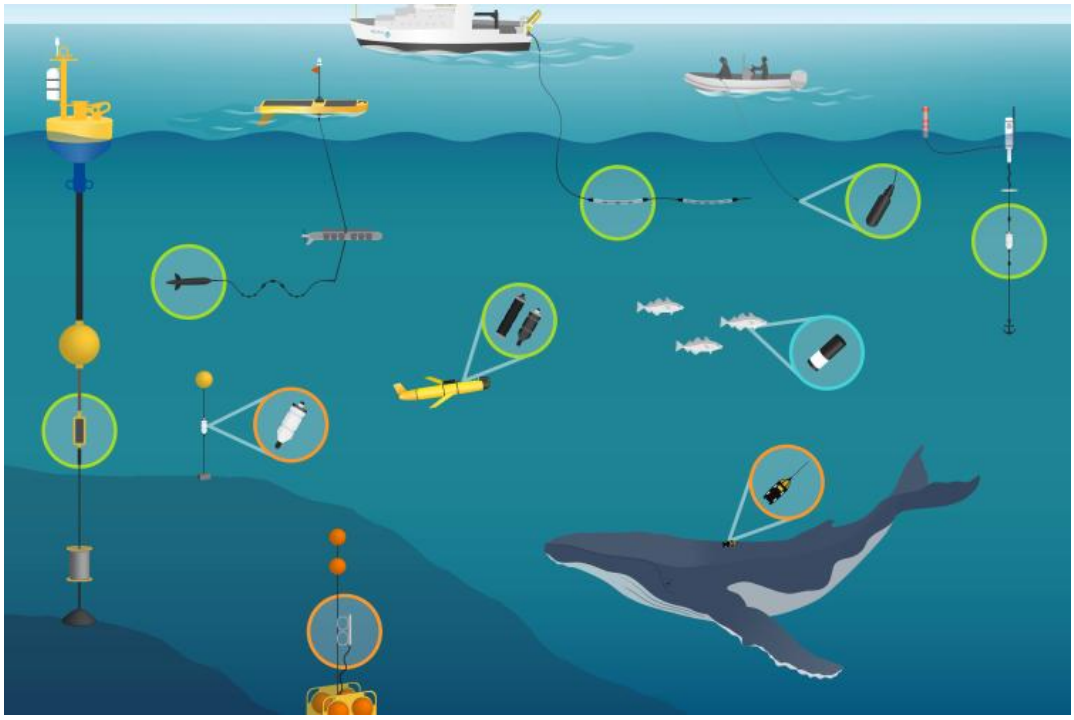
Changing social processes alone cannot control marine debris. Changing government policies, social responsibility, and corporate behavior are vital, but legally binding international treaties are essential for bringing about this change. Despite research and innovation proposing feasible solutions, a circular economic model has yet to be implemented.[41]

2.2 State of the art - whale tracking

Aquatic species evolved to use acoustic signals primarily for communication, navigation, and prey detection since sound propagates more effectively underwater than in the air. Passive acoustic monitoring (PAM) is a method of investigating marine life by recording underwater soundscapes, mammals can be studied for long periods of time with this approach, which is able to continuously monitor mammals and is highly adaptable. Aquatic mammal detection and localization have been made more effective by applying this technique. Scientists have used this technique to study the migration pattern, population structure, and population trends of baleen whales. As an alternative to visual monitoring, PAM records sounds underwater. Normally, visual surveillance is expensive and dependent on fair weather conditions.[13][42]

In the Figure 2.9, passive acoustic technologies are shown as they are used by NOAA Fisheries. These include acoustic tags and bottom-mounted moorings (noted in orange) and drifting buoys, towed arrays, drop hydrophones, gliders, tag receivers, and moored surface buoys (noted in green).[43]

Figure 2.9: Passive acoustic technologies [43]



While PAM can record NARW sounds, it produces a large volume of data that is difficult to analyze manually. To accurately classify these sounds, high-level algorithms are needed. A variety of signal processing methods was commonly used

for the identification and classification of bioacoustic signals, including random forests, regression trees, shallow neural networks, support vector machines (SVM), Gaussian mixture models, etc.[13]

Rasmussen and Sirovic detected and classified fin and blue whale calls from PAM recordings by combining regional convolutional neural network (rCNN) with CNN. For training, they used general spectrograms and two-dimensional (2D) translation and scaling augmentations. CNN architecture was based on a pre-trained ResNet18 network. Both types of calls generated final results of 54% to 57% and 62% to 64%, respectively.[44]

By using Deep Neural Networks (DNN), Padovese et al., 2021 [13] avoided manually labeling years of data collected from numerous hydrophones by leveraging data augmentation on NARW upcalls. Mixup and SpecAugment were applied as augmentation techniques in this study to compare outcomes with and without augmentation. SpecAugment was used on the spectrograms and Mixup was used on the raw waveforms. All three transformations, time-warping, frequency masking, and time masking, were applied simultaneously in the SpecAugment. In terms of precision and recall, Mixup and SpecAugment performed similarly. Magnitude spectrograms were used to train the DNN. The DNN architecture is based on ResNet architecture, which is made up of "six residual blocks with batch normalization and Rectified Linear Units (ReLu)". This layer was fully connected with a discriminating Softmax function that determined if a class score would be positive (call) or negative (no call). In total, 100 epochs were used to train the DNN. It was determined that data augmentation improved DNN's precision to 90.1% and 90.7%, and that the real data is not replaced by data augmentation.

In Vickers' er al., 2021 [45] gunshots detection and NARW upcalls denoising, deep learning was applied. Two approaches were used for denoising the spectrograms before classification: denoising autoencoders and denoising CNNs. A power spectrum-based representation has been used to extract the features, and the CNN has three convolutional layers and two dense layers with the activation functions ReLu and softmax. In this study, the model was trained over 200 iterations for 10 times, and the accuracy is the weighted average of those iterations. There were 17 layers in the residual mapping model. A denoising autoencoder method with higher accuracies than any training without denoising has been reported at 85.18% and CNN at 84.71%.

Esfahanian et al., 2017 [46] analyzed NARW upcalls in two stages, using a detection algorithm for energy and a classifier for binary classification. Following spectrogram creation and normalisation, the binary images are converted to continuous objects in the contour-based approach. In the course of detecting these objects, TFP-2 features are extracted, which include frequency, frequency band, perimeter, area, orientation, and time duration. The texture-based approach involves normalising and equalising the spectrograms, and then extracting features

by means of a Local Binary Pattern (LBP) that describes the patterns of the texture in an image. Linear Discriminant Analysis (LDA) with 91-92% followed by TreeBagger gave the highest classification accuracy with the TFP-2 feature. With only the LBP features in mind, LDA, TreeBagger, and linear SVMs achieve up to 93% accuracy.

In a study conducted by Shiu et al., 2020 [47], several neural networks were tested for automatic recognition of NARW calls. Researchers compared deep neural networks (DNN) with traditional detection algorithms. Specifically, LeNet, BirdNET, VGG and ResNet were used as DNNs, while machine learning (ML) techniques used were shallow neural networks, generalized likelihood ratio tests, multivariate discriminant analysis, decision trees, and boosting classifiers. While the DNNs differed in their classification, they all produced a greater level of accuracy and less false positives than the ML algorithms. LeNet had the highest accuracy and lowest false positive rate, followed by BirdNET, which had similar results but a higher computational cost due to its complex architecture. DNN retrieved 85% to 90% of the upcalls using ML algorithms, while ML algorithms detected 65% of the upcalls.

Skarsoulis et al. [48] devised a method that enables sperm whale clicks to be detected and localised in real time. In this study, three hydrophones were used to detect and localise sounds, which were then sent to a land-based analysis center. The combined data allowed the scientists to recognize and three-dimensionally (3D) locate the animals in real time (every three minutes). Based on peaks in the histograms of arrival-time differences at each hydrophone, an algorithm was applied to identify the dominant separations within the range of regular sperm whale clicks (0.5-2s). Based on patterns of frequent clicks, the arrival time data is then further analysed and used to determine a localisation. Hydrophone separation was large, providing a higher level of localisation accuracy, which proved to be a reliable real-time monitoring tool. As such, it is considered a valuable tool when trying to avoid vessel collisions.

Chapter 3

Theory

3.1 Audio signal processing

Analysing and synthesising periodic signals involves the use of Fourier series; only one time frame of the signal is examined, and only multiples of the signal repetition frequency are considered. "Discrete Fourier transform (DFT) is the orthogonal transform in which reference oscillatory signals are complex-value Fourier harmonics."[49]

In computing DFT, Fast Fourier Transform (FFT) reduces the number of redundant matrix element multiplications with the same signal samples by 100 or more times.[49]

Human speech has a constantly changing frequency content, and an analysed signal can vary in time and frequency content. Consequently, a single spectrum computed for the entire time-varying signal may be false. Using the Short-Time Fourier Transform (STFT), we can track changes in frequency of the signal over time. The STFT step by step process is as follows: splits the signal into numerous overlapping fragments using any window, calculates the FFT spectrum for each fragment and combines them into one matrix, displays the matrix values as a colour or gray-scale image, and monitors changes in the signal spectrum over time - frequency and amplitude modulation curves for individual signal components.[49]

3.1.1 Spectrogram generation

In recent decades, computational bioacoustics has thrived with the development of computer-based sound recording equipment and the expansion of digital technologies such as machine learning, signal processing, and big data. Process methods derive from natural language processing (NLP) and image processing, which are more researched areas of deep learning (DL). Speech analysis and music analysis,

on the other hand, tackle data characteristics, demands, and tasks quite differently than those addressing bioacoustics. Various acoustic signals can help resolve unsolved problems and tasks, but most of them remain unexamined. According to recent studies, bioacoustic classification is based on the following approach. In general, spectrograms are typically divided into fixed sizes. This allows a whole batch of spectrograms to fit easily into the GPU memory. There are three types of spectrograms: standard (linear-frequency), log-frequency and mel. In terms of which spectrogram format is best, no firm recommendations can be made.[50]

Audio classification of raw sound samples is a very challenging task. Therefore, in this project, the frequency and time domains of time series were transformed to create spectrograms based on FFT and STFT. A spectrogram is a tool used for analyzing audio spectral information. Speech recognition relies heavily on this tool. Spectrograms are intensity plots of the magnitudes of the STFT. This represents the audio data in a fundamental way since human hearing depends on a real-time spectrogram encoded by the cochlea (inner ear). Throughout the development of sound synthesis algorithms, spectrogram has been broadly utilised.[51][52].

The sound clips needed to be loaded before spectrograms could be created. Preprocessing of the audio files was achieved by using Python's package *librosa*. *librosa* is a package developed for the analysis of sound and musical signals. As its core functionality, *librosa* enables "audio and time-series operations, spectrogram calculation, time and frequency conversion, and pitch operations.".[51][53]

Data augmentation techniques involve taking a sample of data and making small, irrelevant changes to it to artificially increase the size of the dataset (generally the training set). Audio processing can involve adding low-amplitude noise, combining audio files, time-shifting or even more complex ones such as time or frequency warping.[50]

3.2 Transfer learning - pretrained Neural Networks

Signal and image processing, bioinformatics, and other fields can benefit from machine learning (ML) methods such as deep learning. Convolutional neural networks (CNNs), recurrent neural networks (RNN), reinforcement learning (RL) and others are examples of DL, which can be supervised, semi-supervised, and unsupervised. Typical neural networks (NNs) include a large number of neurons or simple linked processors, each of which produces an activation sequence. Activations are obtained from weighted connections of previously active neurons and from sensors that perceive the environment. In order to implement the desired behavior, the neuronal network learns the location of weights. Because neurons are connected in such a way, the previously stated behavior might require heavy computation steps, and DL should correctly credit these steps.[54].

In this project, classification was achieved through supervised learning. Super-

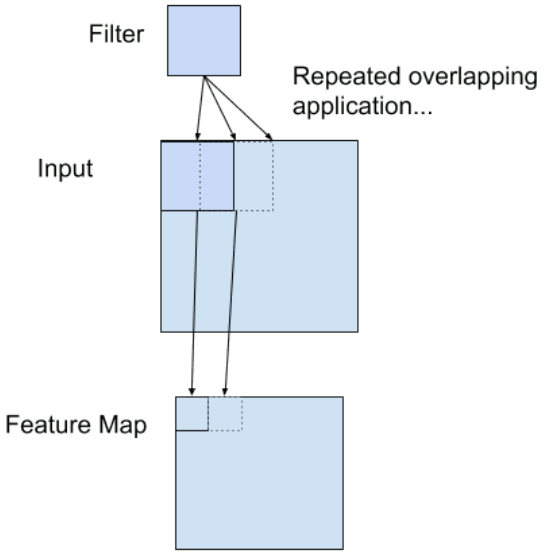
vised learning is a process in which data that has been labeled is used to train the model, and it makes predictions about all points that are not seen. Regression, classification, and ranking problems are usually associated with this approach.[55]

To accomplish the goals of this project, a ML technique called Transfer Learning was applied. Transfer of learning occurs when a model developed for one learning task is applied to another learning task. It is regarded as a research problem in ML, which is concerned with storing and applying knowledge acquired from one problem to another. Essentially, we are using the pre-trained NN from Task1 to reduce the training time in Task2.[56]

CNNs are specialised for handling 2D data. A CNN gradually transfers a convolutional unit's receptive field over a 2D array of input values (e.g., pixels in an image) called a tensor. Having a smaller filter than the input allows the same collection of weights to be multiplied by the input array many times at different points of the input. In particular, the filtering is applied to each filter-sized area of the input data in a systematically organised manner (top-bottom, left-right). Whenever a filter is applied to an input image in order to find a particular feature, there is a chance that the filter will find that feature somewhere in the image. This is called translation invariance. Input filtering is represented by a 2D output array known as a feature map. These features maps are further applied to other layers.[54][57]

Figure 3.1 illustrates how a 2D input can be filtered to create a feature map.

Figure 3.1: Feature map [58]



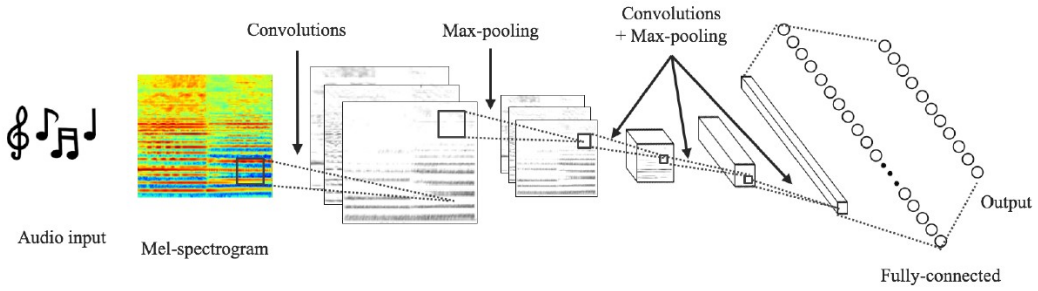
CNNs consist not only of convolutional layers made up of filters and feature

maps, but also of pooling and fully connected or dense layers. Feature maps are downsampled in the pooling layers to maintain an adequate size as the number of features increases. For instance, if we want to classify images of different sizes, we must have a fixed input size for the classification layer. The purpose is to enable following convolution layers to see a larger spatial area of the inputs. Normally, CNNs are concluded with a Flatten operation or a global pooling layer, which converts spatial feature maps into vectors and is followed by Dense layers for classification or regression.[57] [59]

The ability of CNNs to discriminate between images in image recognition tasks made them useful for data analysis.[13]

Figure 3.2 describes an instrument sound recognition example using CNN architecture.

Figure 3.2: The architecture of a CNN [60]

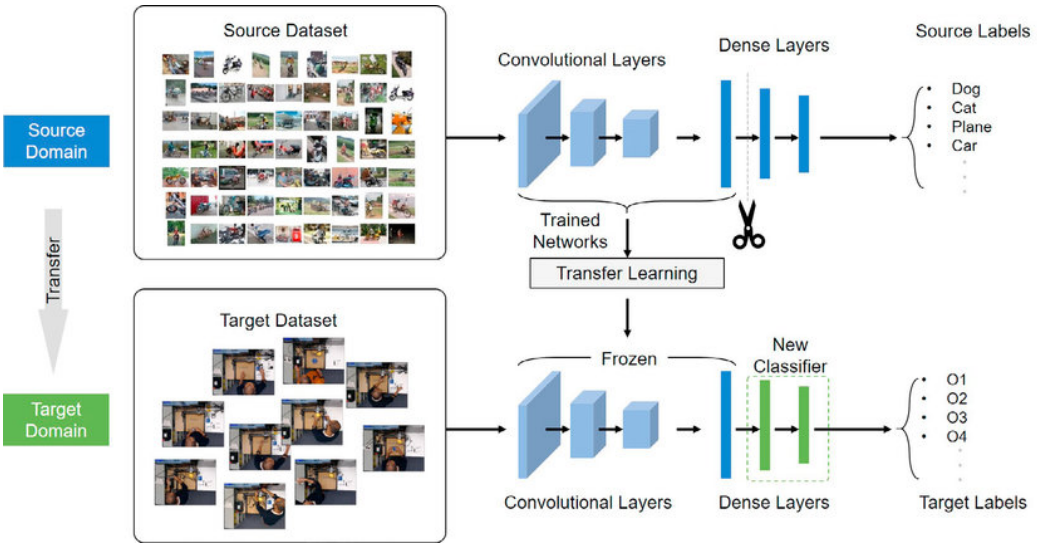


A pre-trained NN, Google's MobileNetV2, was used to enforce Transfer learning. In this layer model, the residual is inverted with a linear bottleneck. Taking a low-dimensional compressed representation as input, it is expanded into a high-dimensional representation, which is then filtered using a lightweight depthwise convolution. In a subsequent step, features are reprojected into a low-dimensional representation using a linear convolution. Although this network retains the design and simplicity of MobileNetV1, it improves the accuracy and delivers state-of-the-art performance on several image classification and mobile detection tasks.[61]

A Transfer learning architecture is shown in Figure 3.3. Training the source dataset involved a great deal of annotated data. When a model is trained, a part of its architecture and weights are frozen and transferred to a target domain. The target model needs another classifier, usually a stack of dense layers, to adapt the source model to the target labels. This means that the input layer is the output layer of the transferred model, whereas the output is set based on the target labels.[62]

TensorFlow interface has been used to implement the MobileNetV2 model. As well as training and inference algorithms for DNN models, TensorFlow has been used for an extensive range of algorithms. In addition, it has been applied to research and the deployment of ML systems across numerous domains such as com-

Figure 3.3: An illustration of the Transfer learning architecture [62]



putational drug discovery, robotics, natural language processing (NLP), computer vision, geographic information extraction, and computer vision.[63]

Chapter 4

Methods

4.1 Machine Learning tools and libraries

The following ML tools and libraries were used in this project. One Jupyter Notebook contained all the code for a test batch trained on Aalborg University's ML workstation. The language of choice is Python, and it was used together with the Jupyter notebook for uploading, preprocessing data, creating spectrograms, creating and running the model, and visualizing results.

There are several Python libraries used, including *os*, *tensorflow*, *shutil*, *keras*, *pandas*, *numpy*, *matplotlib*, *scipy*, *librosa* and other. Python, and its libraries, are primarily open source. This means they are supported by a larger development community, which is one reason they were selected for data training and graph visualisation.

Initially, batches were trained on the ML workstation, followed by access being granted to the CLAAUDIA's AI cloud training at Aalborg University. CLAAUDIA utilizes the Slurm queue system and the Singularity container framework.[64]

Supervised (Transfer) learning was selected because the data used for training was labeled, and pre-trained networks offer high accuracy and speed. MobileNetV2 was used to create the base model. Among the reasons for selecting MobileNetV2 is the fact that it is pre-trained using ImageNet data, which contains 1.4 million images and 1000 classes. Because of its accuracy, smaller size, and lower computational load, it was chosen over other pre-trained networks such as GoogleNet and VGG16.[61][65][66]

TensorFlow is used to load and implement the model. It is a suitable choice for research because of its flexibility and it simplifies real-world use of ML by combining all the necessary features in one place, such as high performance and robustness.[63]

4.2 Dataset and preprocessing

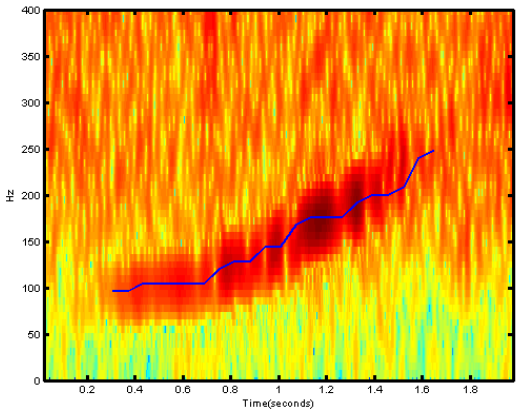
Kaggle was used to download a dataset *"The Marinexplore and Cornell University Whale Detection Challenge"* generated by Cornell University.[67] Google LLC is behind Kaggle, which represents a network of data science enthusiasts. Kaggle allows users to search public datasets, view coding examples, and participate in challenges centered around Data science and ML.[68]

Bioacoustics Research Program at Cornell organized this contest in collaboration with Marinexplore. The competition was aimed at classifying calls made by NARW's.[67]

Data was collected using passive acoustic monitoring (PAM), which has proven to be one of the most effective ways to record marine mammals. A significant challenge of PAM is that it requires a lengthy period of acoustic recordings to distinguish between migration patterns and seasonal behaviors. Background noise often varies significantly among recordings, making analysis difficult.[69]

A clip example of a NARW's call can be seen in Figure 4.1.

Figure 4.1: NARW's call spectrogram example[67]



In total, 30.000 training samples and 54.503 test samples are included in the dataset. The sound clips have a sample rate of 2kHz and each is 2 seconds long. In order to annotate the training set, a *train.csv* file was created, in which labels containing the call were assigned 1, while those without were assigned 0.[67] Due to Supervised learning, only the labeled train dataset was used for training.

To begin with, the implementation was carried out using the ML workstation and a Jupyter notebook. Following the grant access for CLAAUDIA, the notebook was separated into several *.py* files, see Appendix A:

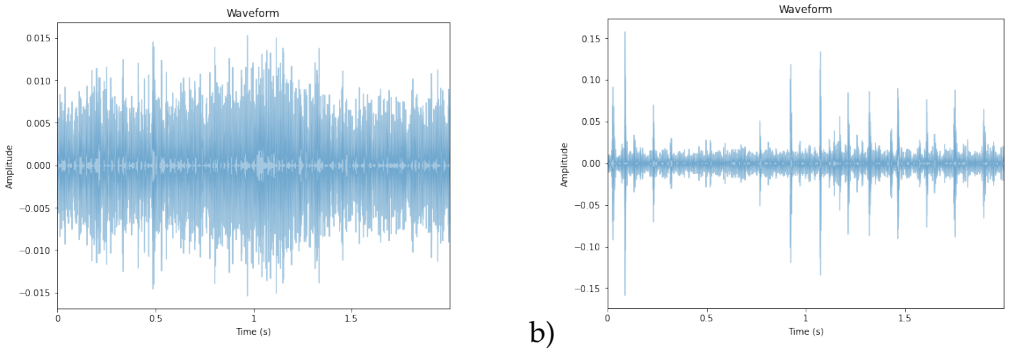
- *plots_raw_audio_spectrograms.py*

- *mfcc_spectrogram_generation1.py*
- *spectrogram_generation2_upgraded.py*
- *class_separation.py*
- *dataset_split.py*
- *mobilenetv2_model.py*

librosa was used to analyze sound, and it was chosen because it is straightforward to use, is fully open-source, and includes a variety of spectral representations.[51][53].

In the *mfcc_spectrogram_generation1* file, the train and test folders have been unpacked and all the required libraries imported, and the code from *plots_raw_audio_spectrograms.py* was used to generate Mel-frequency cepstral coefficients spectrograms (MFCCs).[51][70] MFCCs were selected due to their widespread use in representing audio signals and their ability to provide a basic idea of human frequency perception.[45][53][60] The audio paths have been set, and the size of the figures has been determined. A function for generating spectrograms was defined, small signal frames are spaced apart with a *hop_length* of 128, the windowed signal has a length *n_fft* of 2048, and the variable for creating the spectrogram was created.[53] There is a setting of 2000 Hz for *sample_rate* as determined in the challenge [67]. Every sound clip in the train folder had its MFCC spectrogram plotted as an image with the appropriate name. The newly created spectrograms were saved in the folder *train_mfccs*.

Figure 4.2 shows two waveforms before they are transformed into spectrograms.



a) b)

Figure 4.2: Two waveforms - a) NARW upcall is featured in audio clip train55, b) NARW upcall is not present in audio clip train16 [51]

Examples MFCC spectrograms are shown in Figure 4.3.

For initial test training, 4.310 samples were used to train the model for 10, 30 and 50 epochs, and 2.064 samples were used for validation. For the second test

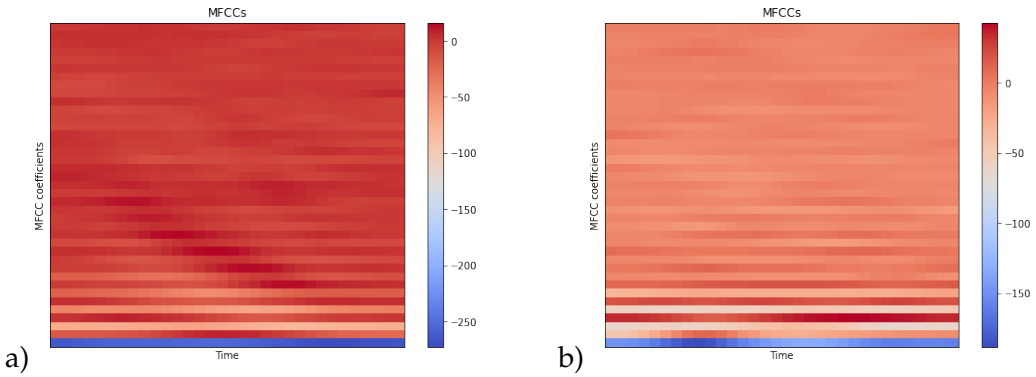


Figure 4.3: Two MFCC spectrograms - a) NARW upcall is featured in audio clip train55, b) NARW upcall is not present in audio clip train16

training 7.476 samples were used for training for 10, 50 and 70 epochs, and 1.870 were used for validation. Both test trainings were conducted without any Data augmentation techniques or fine tuning. Following that, the entire 30.000 samples were used and trained for up to 20 epochs. MFCCs were fed into a model and trained, the results are discussed in Chapter 5 Results and discussion.

As a result of further research, additional information was discovered, so new spectrograms had to be made to get even better results. Stowell [50] notes that MFCC spectrograms have been widely used in previous eras of acoustic analysis. As a result, they are probably not a suitable match to CNN architectures since sounds are not typically shift-invariant along the MFCC coefficient axis. Based on DL assessments, MFCCs are found to underperform less-preprocessed representations like the mel spectrogram.

Magnitude spectrograms were chosen since they are used so frequently in research. By utilizing spectrograms, DL makes use of diverse information, meaning that the input can be compared to digital images. Since mel spectrograms are less suitable for non-human data, they were omitted.[50]

New standard spectrograms were calculated with *hop_length* and *n_fft* remaining the same, and the STFT was applied to get magnitude by calculating absolute values on complex numbers.

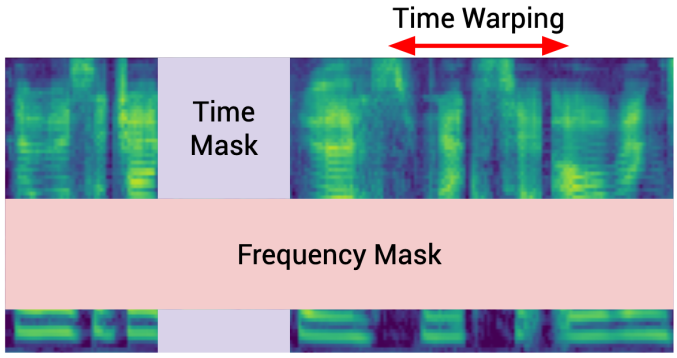
4.3 Data augmentation

As supervised learning on labeled data often results in significant overfitting, it made sense to try and reduce overfitting through Data augmentation.[57] Through Data augmentation, training datasets were augmented by generating new labeled data based on currently available datasets.[13] First augmentation technique was implemented on MFCC spectrograms. MFCCs were augmented with random ro-

tation, flipping, and width manipulation.

As a second augmentation technique, offline augmentation was applied directly to the creation of spectrograms. Known as SpecAugment, it has been used in speech recognition. Using this method does not require any additional data, so it is simple and compute-efficient to apply. SpecAugment contains three types of deformations of the log mel spectrum. The first one is time-warping, in which the time-series is stretched in the time direction. Another method is frequency masking, where each frequency channel is blocked at random using a mask. Thirdly, there is time masking, which masks a set of consecutive time steps.[71][72] Below is an example of how SpecAugment can be combined (Figure 4.4).

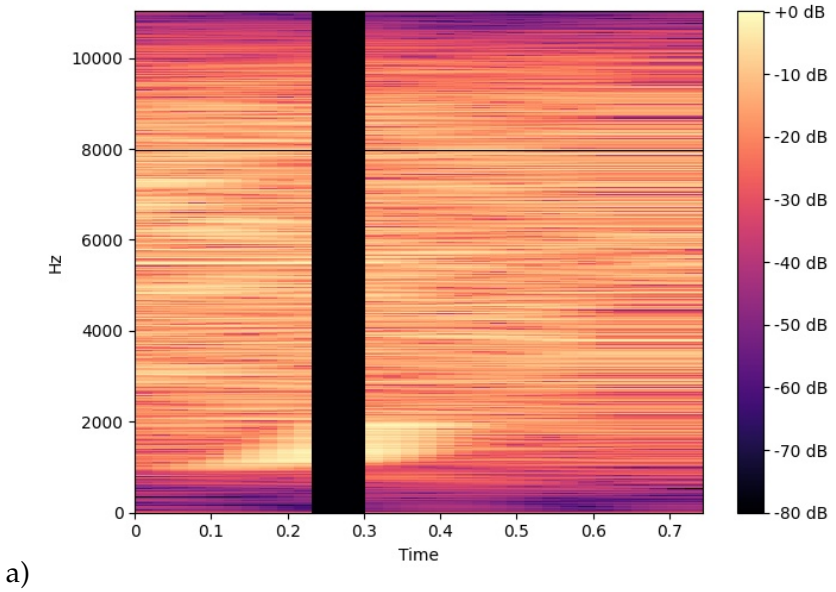
Figure 4.4: SpecAugment example [73]



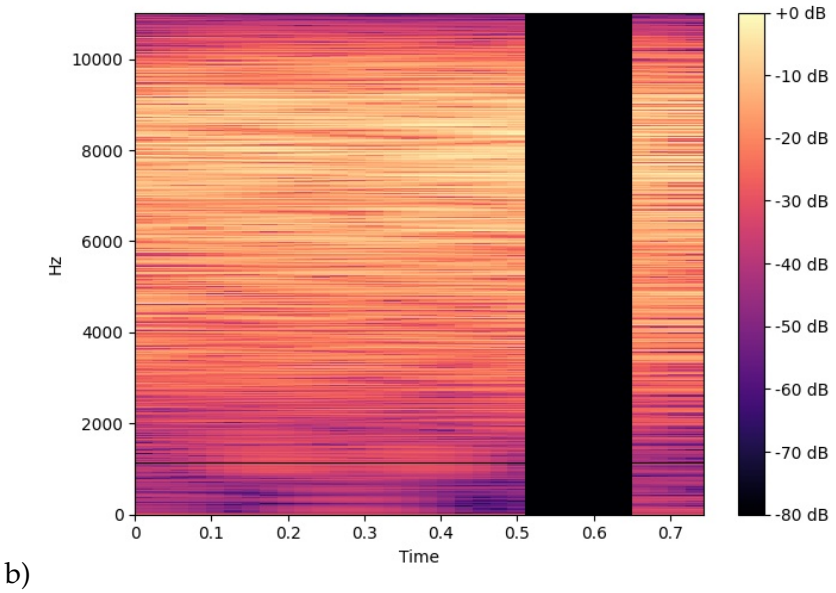
As suggested by Padovese et al. [13], all three transformations were applied simultaneously. After augmenting the entire dataset, the final training dataset consisted of 60.000 samples. The time warping (*sparse_image_warp* method) parameter was set to 8, frequency masking parameter to 7 and time masking parameter to 5, time and frequency maskings were applied six times;
spectrogram_generation2_upgraded.py. As shown in Figure 4.5, spectrogram examples augmented by SpecAugment are depicted. In addition, new labels for the new data were created as well.[74][75]

Once again, spectrograms were created, augmented and saved as images in the train folder. Then, in *class_separation.py*, the corresponding libraries were loaded, labels were loaded, and the train folder was separated into two class folders (0 and 1). Sounds in folder 0 were classified as noise, and those in folder 1 were classified as NARW calls. The train folder was then divided into folders for training and validation in the ratio 80:20 (*dataset_split.py*), which resulted in 47.999 training images and 12.001 validation images.[66]

In *mobilenetv2_model.py*, all the libraries and labels were loaded, then the train folder and the validation folder were loaded into the TensorFlow source dataset, which generated batches of images from the subfolders of each class, together with their labels. Following that, a test set is created from the validation set, and images



a)



b)

Figure 4.5: SpecAugmented spectrograms - a) NARW upcall is featured in audio clip augtrain23209, b) NARW upcall is not present in audio clip augtrain18552

are retrieved by buffered prefetching from the disk. The following is the data augmentation for MFCC spectrograms, which was not used with the pre-augmented data. Lastly, pixel values are rescaled as part of the input preprocessing.[66]

4.4 Machine Learning model

Following preprocessing, the base model was created based on the pretrained MobileNetV2 model. Initially, the model was instantiated and preloaded with weights, but it did not include the layers at the top, since that is the most optimal situation for feature extraction. After the feature extractor performed its work, each image became a block of features with dimensions of 5*5*1280 and the convolutional base was frozen. Through freezing, the weights in a given layer were not updated when the layer was training. The features were subsequently combined into one prediction per image.[66]

A model was built using the Keras Functional API by combining the (data augmentation), rescaling, base model, and feature extractor layers. The Adam optimization algorithm was applied for each weight with a learning rate of 0.001 or 0.0001 as it adapts individually as it learns, and obtains satisfactory results quickly. The initial number of epochs was set after the model was compiled. It took approximately 25 hours to train the model for 20 epochs. The results of the training were plotted and saved, and the evaluation of the model produced performance metrics for overall validation loss and accuracy.[66][76]

A method of fine tuning was used to improve the accuracy of the model. The weights of the top layers of the pre-trained model were applied to the already trained classifier. Using this method, it was possible to tune the weights from generic feature maps to features specific to this dataset. The base model was unfrozen and the bottom layers were set to untrainable. Again the model was trained for a certain number of epochs. Following training, the new data were used to verify the training using the test set and the results were shown in the graph. Performance metrics were also displayed.[66]

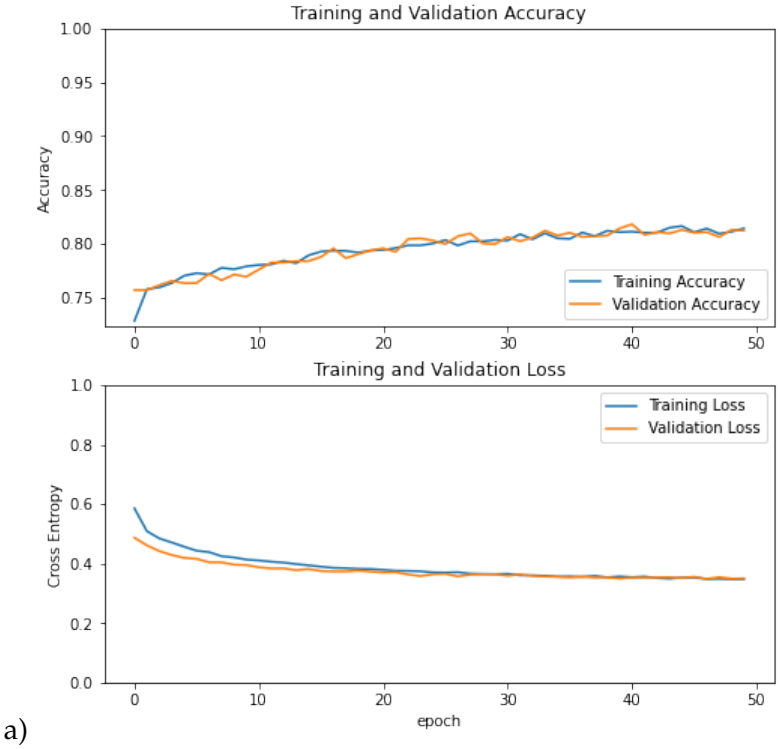
Following is more information about the changes and the comparison of the model results before and after data augmentation; Chapter 5 Results and discussion.

Chapter 5

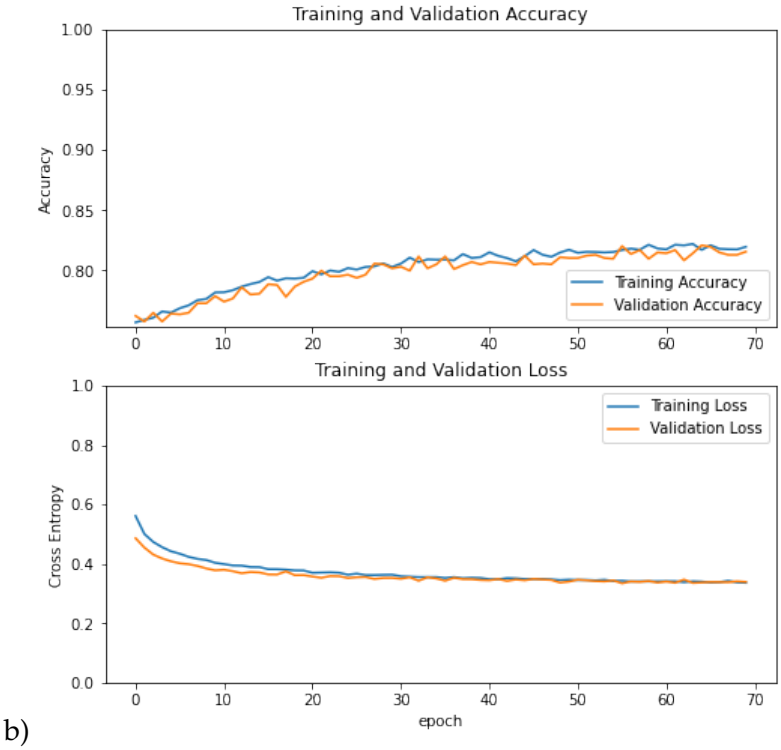
Results and discussion

The first trainings were performed using MFCCs spectrograms. The The smallest data batch trained for 50 epochs resulted in a final accuracy of 75% as presented in Figure 5.1 a). Due to poorer training results, the proposed data augmentation was used only on the first training. When trained for 70 epochs with a slightly bigger data batch and no data augmentation or fine tuning, accuracy reached 82% as depicted in Figure 5.1 b).[66] This was accomplished by setting the *BATCH_SIZE* parameter to 32 and the *base_learning_rate* parameter to 0.0001. Overfitting is apparent in both graphs.[57]

An illustration of the training of the whole dataset for 10 epochs is shown in Figure 5.2. Accordingly, the final accuracy is 81%, signaling a more stable training process, however underfitting or indicating a problem with the data.



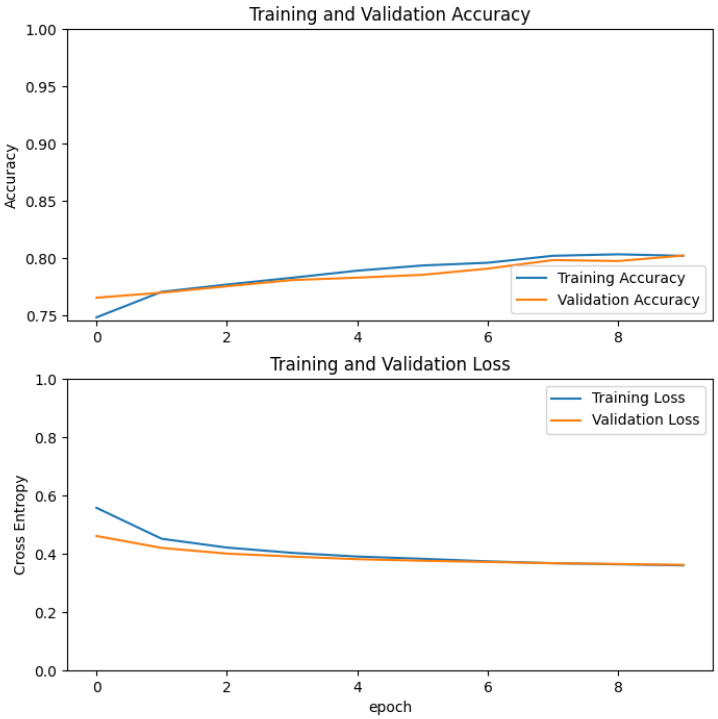
a)



b)

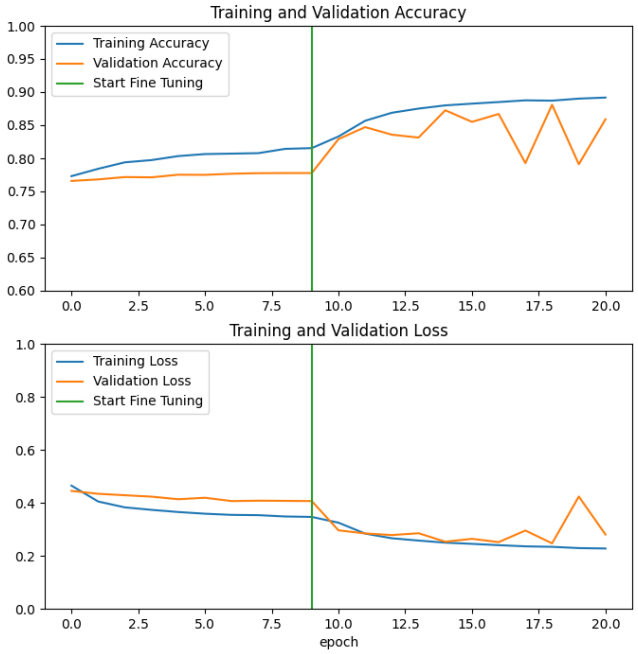
Figure 5.1: Small batch dataset performance through - a) 50 epochs, b) 70 epochs

Figure 5.2: The entire dataset training through 10 epochs

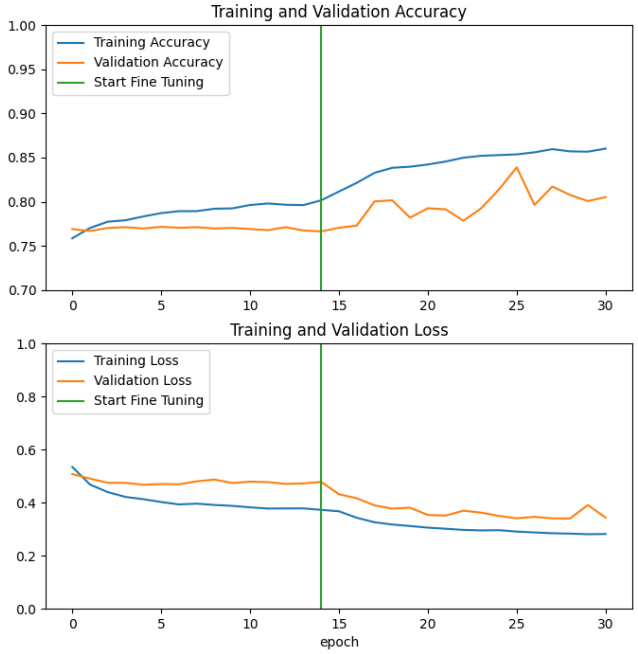


5.1 Results with Data augmentation

A variety of data augmentation and fine-tuning effects were examined in order to improve training results and eliminate underfitting. The techniques involved in data augmentation were part of the TensorFlow platform. 154 layers were included in the base model, and fine-tuning was initiated from the 100th layer.[66] In the training displayed in Figure 5.3 a), *BATCH_SIZE* was set to 16 and *RandomWidth(factor=(0.2, 0.3), interpolation='gaussian')* was used in the data augmentation. For the training shown in Figure 5.3 b), the *BATCH_SIZE* parameter was set to 100, the dropout layer frequency rate was set to 0.3, and *RandomFlip('horizontal')* and *RandomRotation(0.2)* layers were used for augmentation. Both Figure 5.3 a) and Figure 5.3 b) exhibit a significantly higher validation loss in the validation set than in the training set throughout the training process. Accordingly, the final accuracy reached was 84% (Figure 5.3 a)) and 79% (Figure 5.3 b)) respectively, with overfitting and high variance characteristic of both models.[57][77]



a)



b)

Figure 5.3: The entire dataset training with fine tuning and data augmentation - a) 20 epochs - *BATCH_SIZE=16* and data augmentation with *RandomWidth(factor=(0.2, 0.3), interpolation='gaussian')*, b) 30 epochs - *BATCH_SIZE=100* and data augmentation with *RandomFlip('horizontal')* and *RandomRotation(0.2)*

Figure 5.4: The entire dataset training with fine tuning and data augmentation through 20 epochs - *BATCH_SIZE=100*, *base_learning_rate=0.002* and *layers.Dropout(0.3)*

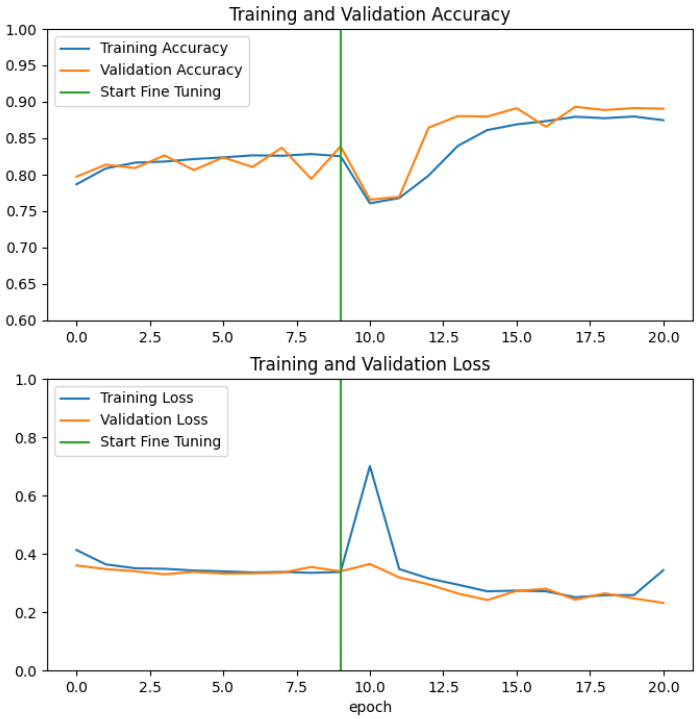
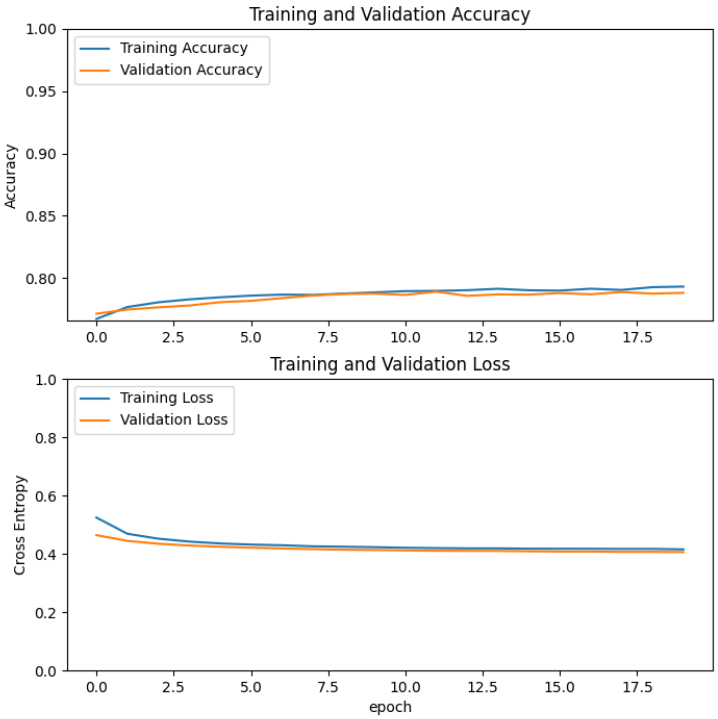


Figure 5.4 depicts the training results where the learning rate is 0.002, the *BATCH_SIZE* is 100, and the dropout layer frequency rate was set to 0.3. Due to an increase in learning rate, a substantial spike in loss is evident once the fine-tuning started. Despite the final accuracy achieved of 88%, the curves show a high degree of variance and overfitting.[77]

Figure 5.5: The entire pre-augmented dataset training through 20 epochs - *BATCH_SIZE=16*



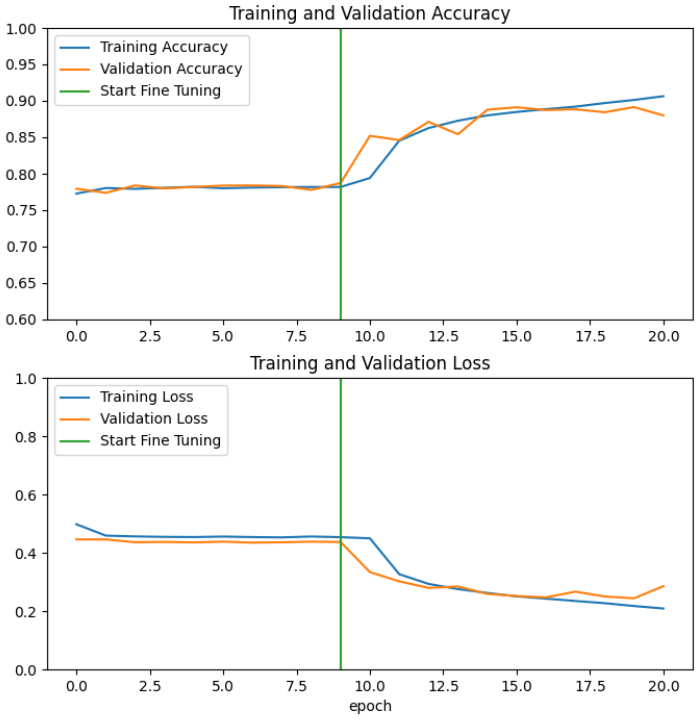
Magnitude spectrograms were made in order to improve the results and reduce the overfitting. After that, SpecAugment was applied to pre-enhance the spectrograms. The data augmentation performed by TensorFlow and the MFCCs spectrograms were directly compared with the Magnitude spectrograms pre-augmented with SpecAugment in another training session. Based on the results shown in Figure C.5, it is evident that pre-augmenting the data with the SpecAugment method enhanced the precision of the model when comparing it to the Figure 5.3 a). Overall, the model had an accuracy of 88%.

As in the Padovese et al. [13] study, the three transformations were applied to the data augmentation of the two classes. Figure 5.5 illustrates the training of the pre-augmented data without fine tuning over 20 epochs and a *BATCH_SIZE* of 16. It was concluded that the final accuracy reached was 79%, and despite the accuracy in Figure 5.2) being higher than that in Figure 5.5, further underfitting is apparent [57][77].

Several of the parameters used by Padovese et al. [13] were also employed in this study. For instance, the *base_learning_rate* was set to 0.001, the *BATCH_SIZE* to 128 and the *kernel_regularizer l2*, otherwise known as the weight decay, to 0.01.

Model training was performed for 10 epochs, followed by 10 epochs of fine-tuning. Figure 5.6 shows the training results, which indicate an accuracy of 88%.

Figure 5.6: The entire pre-augmented dataset training through 20 epochs - *BATCH_SIZE=128*, *base_learning_rate=0.001* and *kernel_regularizer=regularizers.l2(0.01)*



The *base_learning_rate* and *kernel_regularizer l2* parameters were tweaked with different values and were trained for a total of 20 epochs. Figure 5.7 shows the accuracy of 87% achieved using a learning rate of 0.0001 and a regularizer of 0.001.

Figure 5.7: The entire pre-augmented dataset training through 20 epochs - *BATCH_SIZE=128, base_learning_rate=0.0001* and *kernel_regularizer=regularizers.l2(0.001)*

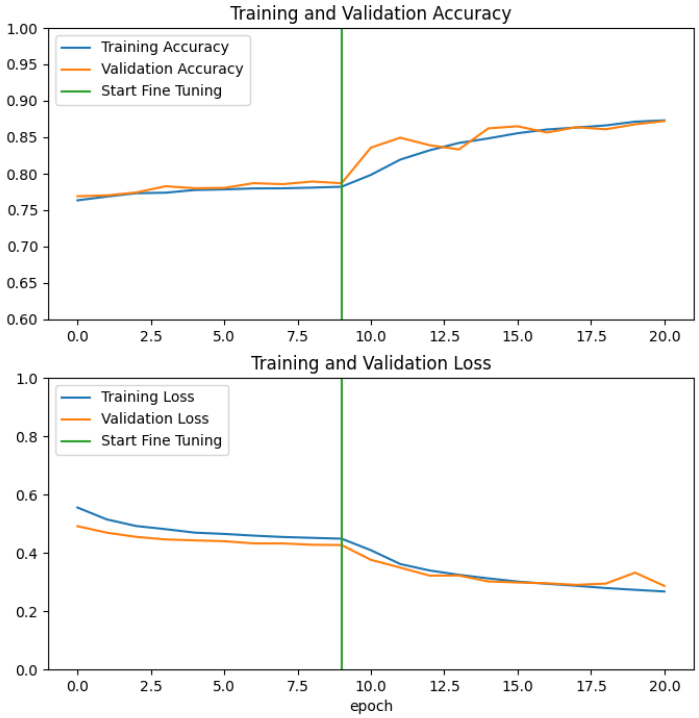
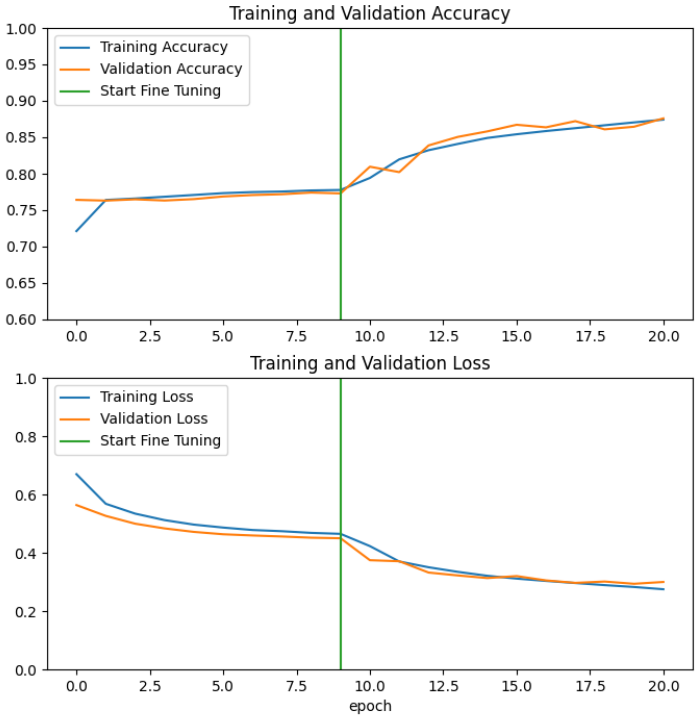


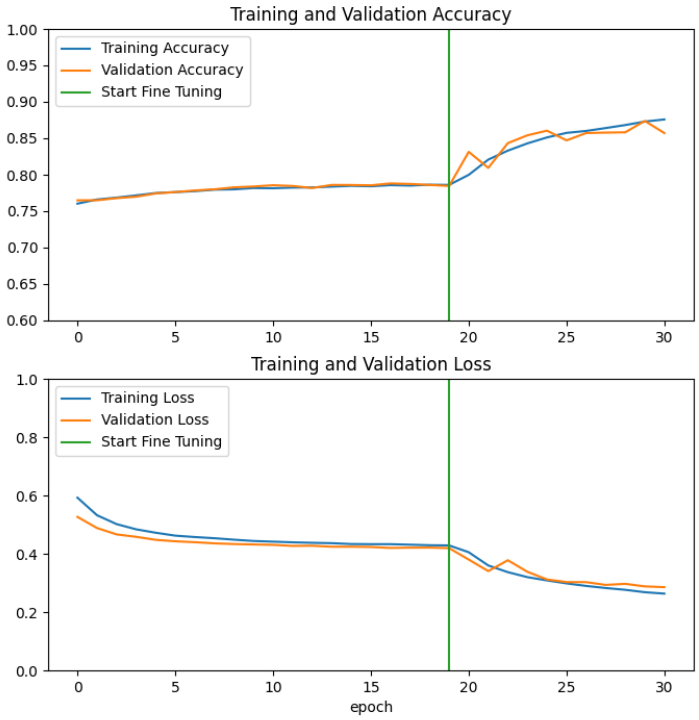
Figure 5.8 indicates the training with 87% accuracy where the weight decay has been increased to 0.01, while the learning rate is kept constant.

Figure 5.8: The entire pre-augmented dataset training through 20 epochs - *BATCH_SIZE=128*, *base_learning_rate=0.0001* and *kernel_regularizer=regularizers.l2(0.01)*



Finally, Figure 5.9 depicts training, in which the learning rate remained constant and weight decay was reduced to 0.0001. Through all the trainings presented in Figures 5.6-5.9 it is clear that a slight reduction in overfitting can be achieved by varying the learning rate and the weight decay, as well as by increasing the number of epochs. While the overall accuracy (87%) in Figure 5.9 was reduced by 1% when compared to the results obtained in Figure 5.6, the final loss was reduced by 3%, and the graph shows less overfitting. Due to this tweaked pre-trained model, the highest accuracy score was 88%.

Figure 5.9: The entire pre-augmented dataset training through 30 epochs - *BATCH_SIZE=128*, *base_learning_rate=0.0001* and *kernel_regularizer=regularizers.l2(0.0001)*



In summary, the observation is that the model greatly benefits from the pre-augmented data. With the use of the fine-tuning method, greater accuracy was achieved than in the initial part of the training. Normalizing the model using the L2 regularization penalty was instrumental in improving the stability of the curves and reducing the overfitting. By using the pre-augmentation of data, training results are clearly better and similar to those obtained by Padovese et al. [13]. Pre-augmentation of offline data has shown promising results and further research is required on this topic. The remaining results are presented in Appendix C.

Chapter 6

Conclusion

Oceans are vast, dark bodies of water that hide their secrets just waiting to be discovered. Since I first started this journey, I have been intrigued by animals purely out of curiosity for them. It has now become the inspiration for my career goals to work with animals. My desire to seek solutions that might help to protect these amazing creatures grows as I learn more about marine science and whales as these amazing animals. It is widely acknowledged that we are running out of time to salvage nature. Therefore, it is quite impressive to see one's ideas on a subject come to life and confirm that they can be accomplished with very promising results for further research. With the use of pre-trained machine learning models that have been trained on billions of images, fairly high levels of accuracy can be achieved in recognizing non-human sounds. Over the years, many new innovations have been created to monitor marine life. To test the machine learning model in an actual environment, a connection to a system controlling an electronic device would be required. The electronic device could be mounted at the bottom of the ship (hull) in such a way that it may be able to pick up the sounds of whales in real time. In terms of electrical engineering, of course, there are many specifications that first need to be met, including water resistance, creating a durable electronic casing, additional filtering, and noise reduction. Flow and surface noise are the biggest problems, both of which are difficult to resolve, as Baumgartner et al. [78] explained.

As evidenced in the interview conducted with A.P. Møller – Mærsk A/S, the shipping industry clearly wants to work together with governmental institutions and non-governmental organizations. Educating the masses to understand how crucial the oceans are is the key to solving this problem. With the end of commercial whaling and cooperation between the shipping and fishing industries, governmental agencies, local communities, and non-governmental organizations, we can develop laws, policies and regulations that will protect not only the ocean and its inhabitants, but also people.

The bottom line is that we are trying to reduce our bad impact on the world,

but is it enough? We will find out in the coming years. There is no doubt that the natural disasters like floods and wildfires that happen around the world are the result of a continuous unsustainable human management of the natural environment, and this is something that will become even worse as time goes by. The environment should, therefore, be treated with love and care, just as we would like to be treated, as it is as fragile as we are.

In the words of my favourite oceanographer Sylvia A. Earle: "Knowing is the key to caring, and with caring there is hope that people will be motivated to take positive actions. They might not care even if they know, but they can't care if they are unaware."[79]

Chapter 7

Future work

Although deep learning has been used for more than a decade on audio tasks, particularly speech and music, it is relatively new to bioacoustics. The study of acoustic signals still poses many unsolved challenges despite the evidence for their existence being present, but not yet fully understood.[50] We can design promising solutions by combining the power of marine life monitoring and machine learning to determine the acoustic signals of marine life in the ocean.

When applying semi-supervised learning, both labeled training data and unlabeled test data may be used to solve the problem of overfitting. In addition to adding or trimming noise, shifting audio in time, combining audio files, etc., there are many other data augmentation techniques worth exploring. In bioacoustics, it is also possible to use a "detect and classify" approach, which can potentially reject a large number of negative sound clips during the detection stage, and then train and apply the classifier for finer discrimination in the second step. In addition to training the dataset for 100 or 200 epochs, various pretrained DNN models should be experimented with. According to Shiu et al. [47], the BirdNet network has good performance. Allen et al. [12] have used a slightly altered ResNet-50. Data may also be pre-trained with Google's AudioSet or with VGG-Sound, as both of these are better suited for sound detection and recognition.[50][71]

In order to spread public awareness, it is worthwhile to put some effort into making this application available on the web or as a mobile app. The development of this system and electronic device might be a collaboration between A.P. Møller – Mærsk A/S and other companies in the future.

This research may be extended to other whale species, such as Bermant et al. [80] who used deep learning with Convolutional (CNNs) and Recurrent Neural Networks (RNNs) and transfer learning for the detection and classification of Sperm whale bioacoustics. Mishachandar and Vairamuthu [81] investigated ocean noise by using CNNs and RNNs, and Zhong et al. [82] used Siamese Neural Networks (SNNs) for the detection and classification of blue whale calls.

Chapter 8

Acknowledgements

I am grateful for Mr Aslak Ross' (Mærsk) willingness to dedicate time and insight to this project as well as their dedication to marine animal conservation. I would like to express my gratitude to Mrs Sabah Afzal and Mrs Julija Ivcenko (Mærsk) for making this interview possible. It is with sincere gratitude that I thank Professor Thomas Arildsen for the constant support regarding CLAUDIA. Lastly, I am extremely grateful to my main supervisor Professor George Palamas for his support, ideas, helpful comments, and suggestions on how to make this project a big success.

Bibliography

[1] IWC Scientific Committee. "Report of the IWC Scientific Committee Workshop on Habitat Degradation". In: *Journal of Cetacean Research and Management* 8.May (2006).

[2] The Editors of Encyclopaedia Britannica. *Sir William Henry Flower, British zoologist*. https://www.britannica.com/biography/William-Henry-Flower. 2022.

[3] W. H. Flower. *On whales, past and present, and their probable origin. A discourse*. The Royal Institution of Great Britain, 1883.

[4] D. F. Eschricht et al. *Recent memoirs on the Cetacea*. Ray society - Robert Hardwicke, 1866.

[5] David Cook et al. "Reflections on the ecosystem services of whales and valuing their contribution to human well-being". In: *Ocean and Coastal Management* 186.December 2019 (2020). DOI: 10.1016/j.ocecoaman.2020.105100.

[6] Joe Roman et al. "Whales as marine ecosystem engineers". In: *Frontiers in Ecology and the Environment* 12.7 (2014), pp. 377–385. DOI: 10.1890/130220.

[7] Ralph Chami et al. "Nature's Solution to Climate Change". In: *Finance and Development* 56.December (2019), pp. 34 –38. URL: https://www.imf.org/external/pubs/ft/fandd/2019/12/natures-solution-to-climate-change-chami.htm.

[8] National Oceanic and Atmospheric Administration (NOAA) Fisheries. *Endangered Species Act Threatened Endangered*. https://www.fisheries.noaa.gov/species-directory/threatened-endangered?title=&species_category=54&species_status=any&regions=all&items_per_page=25&sort=. 2022.

[9] Angela Martin and Natalie Barefoot. "Pacific island whales in a changing climate". In: (2017). DOI: 10.13140/RG.2.2.30073.77928.

[10] Vassili Papastavrou. "Turning the tide: 50 years of collaboration for whale and dolphin conservation". In: *WWF* (2019), p. 48. URL: https://www.worldwildlife.org/publications/turning-the-tide-50-years-of-collaboration-for-whale-and-dolphin-conservation.

[11] Mike Barber et al. “A drop in the ocean Closing the gap in ocean climate finance”. In: November (2021).

[12] Ann N. Allen et al. “A Convolutional Neural Network for Automated Detection of Humpback Whale Song in a Diverse, Long-Term Passive Acoustic Dataset”. In: *Frontiers in Marine Science* 8.March (2021). DOI: 10.3389/fmars.2021.607321.

[13] Bruno Padovese et al. “Data augmentation for the classification of North Atlantic right whales upcalls”. In: *The Journal of the Acoustical Society of America* 149.4 (2021), pp. 2520–2530. DOI: 10.1121/10.0004258.

[14] J G Cooke. “Eubalaena glacialis, North Atlantic Right Whale. The IUCN Red List of Threated Species.” In: 8235 (2020). URL: https://dx.doi.org/10.2305/IUCN.UK.2020-.

[15] Philip K. Hamilton et al. “Genetic identifications challenge our assumptions of physical development and mother–calf associations and separation times: a case study of the North Atlantic right whale (Eubalaena glacialis)”. In: *Mammalian Biology* 0123456789 (2022). DOI: 10.1007/s42991-021-00177-4. URL: https://doi.org/10.1007/s42991-021-00177-4.

[16] Brenna A. Frasier et al. “Genetic examination of historical North Atlantic right whale (Eubalaena glacialis) bone specimens from the eastern North Atlantic: Insights into species history, transoceanic population structure, and genetic diversity”. In: *Marine Mammal Science* March 2021 (2022), pp. 1–20. DOI: 10.1111/mms.12916.

[17] Erin L. Meyer-Gutbrod et al. “Ocean regime shift is driving collapse of the North Atlantic right whale population”. In: *Oceanography* 34.3 (2021), pp. 22–31. DOI: 10.5670/oceanog.2021.308.

[18] Mark Carwardine. *Handbook of Whales, Dolphins and Porpoises*. Bloomsbury Publishing Plc, 2020.

[19] Reviewed by Nick Pyenson The Ocean Portal team. *Whales*. https://ocean.si.edu/ocean-life/marine-mammals/whales. 2022.

[20] The Editors of Encyclopaedia Britannica. *Whaling - human predation*. https://www.britannica.com/topic/whaling. 2022.

[21] Robert C. Rocha, Phillip J. Clapham, and Yulia V. Ivashchenko. “Emptying the oceans: A summary of industrial Whaling catches in the 20th century”. In: *Marine Fisheries Review* 76.4 (2014), pp. 37–48. DOI: 10.7755/MFR.76.4.3.

[22] Daniel Cressey. “World ’s whaling slaughter tallied”. In: *Nature* 519.7542 (2015), pp. 140–141.

[23] The Editors of Encyclopaedia Britannica. *Whale*. https://www.britannica.com/animal/whale. 2022.

[24] Joshua D. Stewart et al. "Decreasing body lengths in North Atlantic right whales". In: *Current Biology* 31.14 (2021), 3174–3179.e3. DOI: 10.1016/j.cub.2021.04.067. URL: https://doi.org/10.1016/j.cub.2021.04.067.

[25] The Editors of Encyclopaedia Britannica. *Right whale*. https://www.britannica.com/animal/right-whale. 2022.

[26] Stephanie Sardelis. *Why do whales sing?* https://ed.ted.com/lessons/how-do-whales-sing-stephanie-sardelis. 2016.

[27] Amalie Rosenkvist et al. "Hearing with eyes in virtual reality". In: *26th IEEE Conference on Virtual Reality and 3D User Interfaces, VR 2019 - Proceedings* December 2019 (2019), pp. 1349–1350. DOI: 10.1109/VR.2019.8797903.

[28] Stephanie Sardelis. *2017–2022 North Atlantic Right Whale Unusual Mortality Event*. https://www.fisheries.noaa.gov/national/marine-life-distress/2017-2022-north-atlantic-right-whale-unusual-mortality-event. 2022.

[29] Loïcka MR Baille and Daniel P Zitterbart. "Effectiveness of surface-based North Atlantic right whale detection methods for vessel strike mitigation". In: (2021), pp. 1–36. URL: https://doi.org/10.1101/2021.08.27.457997.

[30] Renée P. Schoeman, Claire Patterson-Abrolat, and Stephanie Plön. "A Global Review of Vessel Collisions With Marine Animals". In: *Frontiers in Marine Science* 7.May (2020), pp. 1–25. DOI: 10.3389/fmars.2020.00292.

[31] Michael J. Moore et al. "Criteria and case definitions for serious injury and death of pinnipeds and cetaceans caused by anthropogenic trauma". In: *Diseases of Aquatic Organisms* 103.3 (2013), pp. 229–264. DOI: 10.3354/dao02566.

[32] Scott M. Gende et al. "Active whale avoidance by large ships: Components and constraints of a complementary approach to reducing ship strike risk". In: *Frontiers in Marine Science* 6.SEP (2019), pp. 1–19. DOI: 10.3389/fmars.2019.00592.

[33] K. Cates et al. "Strategic plan to mitigate the impacts of ship strikes on cetacean populations: 2017-2020". In: *International Whaling Commission* March 2017 (2017), p. 17.

[34] Heather Breeze et al. "Efforts to advance underwater noise management in Canada: Introduction to the Marine Pollution Bulletin Special Issue". In: *Marine Pollution Bulletin* 178.January (2022), p. 113596. DOI: 10.1016/j.marpolbul.2022.113596.

[35] Joy E. Stanistreet et al. "Changes in the acoustic activity of beaked whales and sperm whales recorded during a naval training exercise off eastern Canada". In: *Scientific Reports* 12.1 (2022), pp. 1–13. DOI: 10.1038/s41598-022-05930-4. URL: https://doi.org/10.1038/s41598-022-05930-4.

[36] Leanna P. Matthews and Susan E. Parks. "An overview of North Atlantic right whale acoustic behavior, hearing capabilities, and responses to sound". In: *Marine Pollution Bulletin* 173.PB (2021), p. 11343. DOI: 10.1016/j.marpolbul.2021.113043. URL: https://doi.org/10.1016/j.marpolbul.2021.113043.

[37] International Maritime Organization. *Introduction to IMO*. https://www.imo.org/en/About/Pages/Default.aspx. 2022.

[38] IMO. "Guidance document for minimizing the risk of ship strikes with cetaceans. MEPC.1/Circ.674. 31 July 2009". In: (2009).

[39] IMO. "Identification and protection of special areas and PSSAs - submitted by IWC; MEPC 69/10/3". In: (2016).

[40] International Whaling Commission. *The International Whaling Commission - IWC*. https://iwc.int/. 2022.

[41] Sonja Mareike Eisfeld-Pierantonio, Nino Pierantonio, and Mark P. Simmonds. "The impact of marine debris on cetaceans with consideration of plastics generated by the COVID-19 pandemic". In: *Environmental Pollution* 300.January (2022), p. 118967. DOI: 10.1016/j.envpol.2022.118967. URL: https://doi.org/10.1016/j.envpol.2022.118967.

[42] Lei Li et al. "Automated classification of Tursiops aduncus whistles based on a depth-wise separable convolutional neural network and data augmentation". In: *The Journal of the Acoustical Society of America* 150.5 (2021), pp. 3861–3873. DOI: 10.1121/10.0007291.

[43] Northeast Fisheries Science Center. *Passive Acoustic Technologies*. https://www.fisheries.noaa.gov/new-england-mid-atlantic/science-data/passive-acoustic-technologies. 2022.

[44] Jeppe Have Rasmussen and Ana Širović. "Automatic detection and classification of baleen whale social calls using convolutional neural networks". In: *The Journal of the Acoustical Society of America* 149.5 (2021), pp. 3635–3644. DOI: 10.1121/10.0005047.

[45] 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. DOI: 10.1121/10.0005128.

[46] Mahdi Esfahanian et al. "Two-stage detection of north Atlantic right whale upcalls using local binary patterns and machine learning algorithms". In: *Applied Acoustics* 120 (2017), pp. 158–166. DOI: 10.1016/j.apacoust.2017.01.025. URL: http://dx.doi.org/10.1016/j.apacoust.2017.01.025.

[47] Yu Shiu et al. "Deep neural networks for automated detection of marine mammal species". In: *Scientific Reports* 10.1 (2020), pp. 1–12. DOI: 10.1038/s41598-020-57549-y.

[48] Emmanuel K Skarsoulis et al. "A Real-Time Acoustic Observatory for Sperm-Whale Localization in the Eastern Mediterranean Sea". In: 9.May (2022), pp. 1–18. DOI: 10.3389/fmars.2022.873888.

[49] Tomasz P. Zieliński. *Starting Digital Signal Processing in Telecommunication Engineering*. Springer Nature Switzerland AG, 2021. URL: http://link.springer.com/10.1007/978-3-030-49256-4.

[50] Dan Stowell. "Computational bioacoustics with deep learning: a review and roadmap". In: *PeerJ* 10 (2022), e13152. DOI: 10.7717/peerj.13152.

[51] Mirjana Erceg. *North Atlantic right whale calls classification with Convolutional Neural Network*. 2021.

[52] Julius O. Smith III. *Mathematics of the Discrete Fourier Transform: With Audio Applications*. 2002, p. 322. URL: https://ccrma.stanford.edu/$\sim$jos/mdft/.

[53] Brian McFee et al. "librosa: Audio and Music Signal Analysis in Python". In: *Proceedings of the 14th Python in Science Conference* Scipy (2015), pp. 18–24. DOI: 10.25080/majora-7b98e3ed-003.

[54] Jürgen Schmidhuber. "Deep Learning in neural networks: An overview". In: *Neural Networks* 61 (2015), pp. 85–117. DOI: 10.1016/j.neunet.2014.09.003. URL: http://dx.doi.org/10.1016/j.neunet.2014.09.003.

[55] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning, second edition*. The MIT Press, 2018, p. 475.

[56] Stevo Bozinovski. "Reminder of the first paper on transfer learning in neural networks, 1976". In: *Informatica (Slovenia)* 44.3 (2020), pp. 291–302. DOI: 10.31449/INF.V44I3.2828.

[57] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. The MIT Press, 2016, p. 800.

[58] Jason Brownlee. *How Do Convolutional Layers Work in Deep Learning Neural Networks?* https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/. 2020.

[59] François Chollet. *Deep Learning with Python*. Manning Publications Co., 2018, p. 353.

[60] Yoonchang Han, Jaehun Kim, and Kyogu Lee. "Deep Convolutional Neural Networks for Predominant Instrument Recognition in Polyphonic Music". In: *IEEE/ACM Transactions on Audio Speech and Language Processing* 25.1 (2017), pp. 208–221. DOI: 10.1109/TASLP.2016.2632307.

[61] Mark Sandler et al. "MobileNetV2: Inverted Residuals and Linear Bottlenecks". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2018), pp. 4510–4520. DOI: 10.1109/CVPR.2018.00474.

[62] Wenjin Tao et al. "Real-time assembly operation recognition with fog computing and transfer learning for human-centered intelligent manufacturing". In: *Procedia Manufacturing* 48 (2020), pp. 926–931. DOI: 10.1016/j.promfg.2020.05.131. URL: https://doi.org/10.1016/j.promfg.2020.05.131.

[63] Martín Abadi et al. "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems". In: (2016). URL: http://arxiv.org/abs/1603.04467.

[64] Thomas Arildsen. *AI Cloud at AAU user information*. https://git.its.aau.dk/CLAAUDIA/docs_aicloud/src/branch/master/aicloud_slurm. 2022.

[65] Andrew G. Howard et al. "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications". In: (2017). URL: http://arxiv.org/abs/1704.04861.

[66] Google Developers. *Transfer learning and fine-tuning*. https://www.tensorflow.org/tutorials/images/transfer_learning. 2022.

[67] Cornell University and Inc. the Cornell Research Foundation. *Real-time Monitoring System for Detecting North Atlantic Right Whales*. https://www.kaggle.com/c/whale-detection-challenge/overview. 2013.

[68] Anthony Goldbloom and Ben Hamner. *Kaggle Your Home for Data Science*. https://www.kaggle.com/. 2022.

[69] Peter J. Dugan et al. "North atlantic right whale acoustic signal processing: Part I. comparison of machine learning recognition algorithms". In: *2010 Long Island Systems, Applications and Technology Conference, LISAT 10* June (2010). DOI: 10.1109/LISAT.2010.5478268.

[70] Ankur Dhuriya. *Simplifying Audio Data: FFT, STFT & MFCC*. https://medium.com/analytics-vidhya/simplifying-audio-data-fft-stft-mfcc-for-machine-learning-and-deep-learning-443a2f962e0e. 2020.

[71] Google Developers. *Audio Data Preparation and Augmentation*. https://www.tensorflow.org/io/tutorials/audio. 2022.

[72] Daniel S. Park et al. "Specaugment: A simple data augmentation method for automatic speech recognition". In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH* 2019-September (2019), pp. 2613–2617. DOI: 10.21437/Interspeech.2019-2680.

[73] Heng-Jui Chang. *Data Augmentation in Automatic Speech Recognition*. https://spectra.mathpix.com/article/2021.09.00002/asr-data-augmentation. 2022.

[74] Demis TaeKyu Eom, Edward J. Yoon, and al. *SpecAugment*. https://github.com/DemisEom/SpecAugment. 2020.

[75] Piyush Vyas. *SpecAugment*. https://github.com/pyyush/SpecAugment. 2020.

[76] Zachary J. Ruff et al. "Automated identification of avian vocalizations with deep convolutional neural networks". In: *Remote Sensing in Ecology and Conservation* 6.1 (2020), pp. 79–92. DOI: 10.1002/rse2.125.

[77] University of Burgos Luis R. Izquierdo. *Machine Learning. Model assessment and model selection*. https://www.youtube.com/watch?v=g5BTk_lywb0&list=PLN4kTzLXGGgWhZw7apsNp_sKxMsnCfzX3&index=1. 2020.

[78] 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." eng. In: *The Journal of the Acoustical Society of America* 149.5 (2021), p. 2950. DOI: 10.1121/10.0004817.

[79] S A Earle and B McKibben. *The world is blue: how our fate and the ocean's are one*. Vol. 47. 08. National Geographic, 2010. DOI: 10.5860/choice.47-4427.

[80] Peter C. Bermant et al. "Deep Machine Learning Techniques for the Detection and Classification of Sperm Whale Bioacoustics". In: *Scientific Reports* 9.1 (2019), pp. 1–10. DOI: 10.1038/s41598-019-48909-4.

[81] B. Mishachandar and S. Vairamuthu. "Diverse ocean noise classification using deep learning". In: *Applied Acoustics* 181 (2021), p. 108141. DOI: 10.1016/j.apacoust.2021.108141. URL: https://doi.org/10.1016/j.apacoust.2021.108141.

[82] "Detecting, classifying, and counting blue whale calls with Siamese neural networks". In: *The Journal of the Acoustical Society of America* 149.5 (2021), pp. 3086–3094. DOI: 10.1121/10.0004828.

[83] Nick Hawkins. *North Atlantic Right whales*. https://nickhawkins.photoshelter.com/gallery/North-Atlantic-Right-whales/G0000gZgmFn3U4Tw/. 2020.

[84] IWC Climate Change Workshop. "Report of the IWC Climate Change Steering". In: December 2021 (2014), pp. 1–49.

Appendix A

Appendix A - Environmental stressors

A.1 Fisheries - bycatch and entanglement

The problem of bycatch, when aquatic life is caught by accident, has implications for conservation and fisheries management around the world. Marine mammals are the most frequently targets, but other animals such as seabirds, turtles, sharks, and other fish are also at risk. Each year, the mortality rate of marine mammals is estimated to be 300.000 worldwide due to bycatch.[40]

Around the world, entanglement is the biggest threat to whales after ship strikes.[9] A lot of cetaceans become entangled in fishing nets, lines, and ropes. As a result, some drown immediately as they cannot reach the surface to breathe. Other survivors die slowly, dragging along their equipment for months or years before they die of starvation or wounds. It is believed that gillnets and entangling nets are most lethal to cetaceans. Monitoring is extremely limited as a result of difficulties evaluating bycatch and an insufficient reporting of data.[40]

Figure 2.9 shows a severely entangled North Atlantic right whale in the Gulf of St. Lawrence, Canada, suffered damage to its baleen due to fishing ropes wrapping around its head and mouth.

Both the severity and frequency of entanglements appear to be on the rise. It appears that NARWs acquire new entanglement scars almost every year, juveniles even more frequently than adults. According to estimates, individuals who are caught in fishing gear die at a rate of about 25% in the first year. Entanglement damages the whale's fitness balance, resulting in poorer body condition, lower survival and lower reproduction, although it may not be fatal in all cases.[14]

Figure A.1: Entangled NARW [83]

A.2 Indigenous and commercial whaling

In recent years, Norway, the Faroe Islands (Denmark), Iceland and Japan continued to hunt whales for commercial purposes. Some countries allow indigenous whaling because it plays an important role in native culture and diet. Some of the IWC member countries conduct indigenous whale hunting, including Denmark (Greenland), the U.S. (Alaska), Russia (Chukokka), as well as St Vincent and the Grenadines (Bequia).[40]

A.3 Climate change and marine debris

Unless CO_2 and other green gas emissions are drastically reduced, 1.5-2°C of global warming will be exceeded in the 21st century. The Antarctic and Greenland ice sheets are melting, the ocean is warming, sea levels are rising, the deep oceans are deoxygenating and they are becoming acidified. This is primarily due to human activity. Global warming over the past century has reached unprecedented levels since the end of the last deglacial period. If global emissions of greenhouse gases continue, the Arctic Ocean is likely to become ice-free before 2050, a condition that might become the new norm by 2100. Changes in climate affect the availability of

whales' prey and its distribution, changing the habitat locations of whales.[84]

Any material that does not belong in the ocean, such as fishing gear fragments, bottles, plastic, rubber, clothing, soda cans, can be classified as debris. Marine organisms are seriously threatened by the accumulation of human-derived debris in the oceans. The amount of plastic entering and accumulating in marine environments is estimated to be substantial every year. Marine debris can persist if it remains attached to an animal without killing it right away. The debris can still cause severe wounds even after separating from the animal. A lot of debris is small enough to be ingested, so it kills the animals from inside causing muscular, organ and skeletal damage. Plastic makes up most of the litter due to low recycling rates, poor waste management, and durability, and so it lingers in the ocean.[31][41]

Appendix B

Appendix B - Code

B.1 plots_raw_audio_spectrograms.py

```
import librosa
import librosa.display
import matplotlib.pyplot as plt
import numpy as np
import os
os.environ['LIBROSA_CACHE_DIR'] = '/tmp/'

# Visualising one audio clip
audio_fpath = "./train_sounds/"
spectrograms_path = "./train_mfccs/"
audio_clips = os.listdir(audio_fpath)

FIG_SIZE = (8,6)
# data sample rate is 2000 Hz, retrieved from the files
signal, sample_rate = librosa.load(audio_fpath,sr = None)
plt.figure(figsize=FIG_SIZE)
librosa.display.waveplot(signal, sample_rate, alpha=0.4)
plt.xlabel("Time (s)")
plt.ylabel("Amplitude")
plt.title("Waveform")
plt.show()

# perform Fourier transform
fft = np.fft.fft(signal)
# # calculate abs values on complex numbers to get magnitude
spectrum = np.abs(fft)
# create frequency variable
f = np.linspace(0, sample_rate, len(spectrum))
# take half of the spectrum and frequency
```

```
left_spectrum = spectrum[:int(len(spectrum)/2)]
left_f = f[:int(len(spectrum)/2)]
# plot spectrum
plt.figure(figsize=FIG_SIZE)
plt.plot(left_f, left_spectrum, alpha=0.4)
plt.xlabel("Frequency")
plt.ylabel("Magnitude")
plt.title("Power spectrum")
plt.show()
# Check, the frequencies go from 0 to FS/2 = 1000.

# STFT -> spectrogram
# Your FS = 2000 Hz, 8 times higher
hop_length = 128 # in num. of samples
n_fft = 2048 # window in num. of samples

# calculate duration hop length and window in seconds
hop_length_duration = float(hop_length)/sample_rate
n_fft_duration = float(n_fft)/sample_rate

print("STFT hop length duration is: {}s".format(
   hop_length_duration))
print("STFT window duration is: {}s".format(n_fft_duration))

# perform stft
stft = librosa.stft(signal, n_fft=n_fft, hop_length=hop_length)

# calculate abs values on complex numbers to get magnitude
spectrogram = np.abs(stft)

# display spectrogram
plt.figure(figsize=FIG_SIZE)
librosa.display.specshow(spectrogram,
                         sr=sample_rate,
                         hop_length=hop_length)
plt.xlabel("Time")
plt.ylabel("Frequency")
plt.colorbar()
plt.title("Spectrogram")
plt.show()

# apply logarithm to cast amplitude to Decibels
log_spectrogram = librosa.amplitude_to_db(spectrogram)
plt.figure(figsize=FIG_SIZE)
librosa.display.specshow(log_spectrogram,
                         sr=sample_rate,
                         hop_length=hop_length)
```

```
plt.xlabel("Time")
plt.ylabel("Frequency")
plt.colorbar(format="%+2.0f dB")
plt.title("Spectrogram (dB)")
plt.show()

# MFCCs
# extract 13 MFCCs
MFCCs = librosa.feature.mfcc(signal,
                             sample_rate,
                             n_fft=n_fft,
                             hop_length=hop_length,
                             n_mfcc=39)
# display MFCCs
plt.figure(figsize=FIG_SIZE)
librosa.display.specshow(MFCCs, sr=sample_rate,
hop_length=hop_length)
plt.xlabel("Time")
plt.ylabel("MFCC coefficients")
plt.colorbar()
plt.title("MFCCs")
plt.show()
```

B.2 mfcc_spectrogram_generation1.py

```
import librosa
import librosa.display
import matplotlib.pyplot as plt
import os
os.environ['LIBROSA_CACHE_DIR'] = '/tmp/'

# Unzipping the sound files from train & test folders
# shutil.unpack_archive("train.zip", "")
# shutil.unpack_archive("test.zip", "")


# GENERATING REGULAR SPECTROGRAMS
audio_fpath = "./train_sounds/"
audio_clips = os.listdir(audio_fpath)
FIG_SIZE = (7, 5)

def generate_spectrogram(signal, sample_rate, save_name):

    hop_length = 128 # in num. of samples
    n_fft = 2048 # window in num. of samples
```

```
    # creating MFCC spectrograms
    mfcc = librosa.feature.mfcc(signal, n_fft=n_fft, hop_length
       =hop_length, n_mfcc=39)

    # plotting the spectrogram
    fig = plt.figure(figsize=FIG_SIZE, dpi=1000, frameon=True)
    ax = fig.add_axes([0, 0, 1, 1], frameon=True)
    ax.axis('on')
    librosa.display.specshow(mfcc, x_axis='time', hop_length=
       hop_length, sr=2000, vmin=-500, vmax=500)
    plt.colorbar()
    plt.ylabel("MFCC coefficients")
    plt.savefig(save_name, pil_kwargs={'quality': 95},
       bbox_inches=0, pad_inches=0)
    librosa.cache.clear()

# Creating sprectrograms for both train and test batch
for i in audio_clips:
    spectrograms_path = "./aug_train_spects/"
    save_name = spectrograms_path + i + ".jpg"
    # check if a file already exists
    if not os.path.exists(save_name):
        signal, sample_rate = librosa.load(audio_fpath + i, sr
           =2000)
        generate_spectrogram(signal, sample_rate, save_name)
        plt.close()
```

B.3 spectrogram_generation2_upgraded.py

```
import librosa
import librosa.display
import matplotlib.pyplot as plt
import numpy as np
import os, sys
os.environ['LIBROSA_CACHE_DIR'] = '/tmp/'
import tensorflow as tf
from sparse_image_warp import sparse_image_warp
from augment import SpecAugment


def sparse_warp(spectrogram, time_warping_para=8):
    """Spec augmentation Calculation Function.

    'SpecAugment' have 3 steps for audio data augmentation.
```

```
    first step is time warping using Tensorflow's
        image_sparse_warp function.
    Second step is frequency masking, last step is time masking
        .

    # Arguments:
      mel_spectrogram(numpy array): audio file path of you want
          to warping and masking.
      time_warping_para(float): Augmentation parameter, "time
         warp parameter W".
        If none, default = 80 for LibriSpeech.

    # Returns
      mel_spectrogram(numpy array): warped and masked mel
         spectrogram.
    """

    fbank_size = tf.shape(input=spectrogram)
    n, v = fbank_size[1], fbank_size[2]

    # Step 1 : Time warping
    # Image warping control point setting.
    # Source
    pt = tf.random.uniform([], time_warping_para, n-
        time_warping_para, tf.int32) # radnom point along the
        time axis
    src_ctr_pt_freq = tf.range(v // 2)  # control points on
        freq-axis
    src_ctr_pt_time = tf.ones_like(src_ctr_pt_freq) * pt  #
        control points on time-axis
    src_ctr_pts = tf.stack((src_ctr_pt_time, src_ctr_pt_freq),
        -1)
    src_ctr_pts = tf.cast(src_ctr_pts, dtype=tf.float32)

    # Destination
    w = tf.random.uniform([], -time_warping_para,
        time_warping_para, tf.int32)  # distance
    dest_ctr_pt_freq = src_ctr_pt_freq
    dest_ctr_pt_time = src_ctr_pt_time + w
    dest_ctr_pts = tf.stack((dest_ctr_pt_time, dest_ctr_pt_freq
        ), -1)
    dest_ctr_pts = tf.cast(dest_ctr_pts, dtype=tf.float32)

    # warp
    source_control_point_locations = tf.expand_dims(src_ctr_pts
        , 0)  # (1, v//2, 2)
    dest_control_point_locations = tf.expand_dims(dest_ctr_pts,
```

```
        0)  # (1, v//2, 2)

    warped_image, _ = sparse_image_warp(spectrogram,
       source_control_point_locations,
       dest_control_point_locations)

    return warped_image


def frequency_masking(spectrogram, v, frequency_masking_para=7,
    frequency_mask_num=2):
    """Spec augmentation Calculation Function.

    'SpecAugment' have 3 steps for audio data augmentation.
    first step is time warping using Tensorflow's
       image_sparse_warp function.
    Second step is frequency masking, last step is time masking
       .

    # Arguments:
      mel_spectrogram(numpy array): audio file path of you want
          to warping and masking.
      frequency_masking_para(float): Augmentation parameter, "
         frequency mask parameter F"
        If none, default = 100 for LibriSpeech.
      frequency_mask_num(float): number of frequency masking
         lines, "m_F".
        If none, default = 1 for LibriSpeech.

    # Returns
      mel_spectrogram(numpy array): warped and masked mel
         spectrogram.
    """
    # Step 2 : Frequency masking
    fbank_size = tf.shape(input=spectrogram)
    n, v = fbank_size[1], fbank_size[2]

    for i in range(frequency_mask_num):
        f = tf.random.uniform([], minval=0, maxval=
           frequency_masking_para, dtype=tf.int32)
        v = tf.cast(v, dtype=tf.int32)
        f0 = tf.random.uniform([], minval=0, maxval=v-f, dtype=
           tf.int32)

        # warped_mel_spectrogram[f0:f0 + f, :] = 0
        mask = tf.concat((tf.ones(shape=(1, n, v - f0 - f, 1)),
                          tf.zeros(shape=(1, n, f, 1)),
```

```
                        tf.ones(shape=(1, n, f0, 1)),
                        ), 2)
        spectrogram = spectrogram * mask
    return tf.cast(spectrogram, dtype=tf.float32)


def time_masking(spectrogram, tau, time_masking_para=5,
    time_mask_num=4):
    """Spec augmentation Calculation Function.

    'SpecAugment' have 3 steps for audio data augmentation.
    first step is time warping using Tensorflow's
       image_sparse_warp function.
    Second step is frequency masking, last step is time masking
       .

    # Arguments:
      mel_spectrogram(numpy array): audio file path of you want
          to warping and masking.
      time_masking_para(float): Augmentation parameter, "time
         mask parameter T"
        If none, default = 27 for LibriSpeech.
      time_mask_num(float): number of time masking lines, "m_T
         ".
        If none, default = 1 for LibriSpeech.

    # Returns
      mel_spectrogram(numpy array): warped and masked mel
         spectrogram.
    """
    fbank_size = tf.shape(input=spectrogram)
    n, v = fbank_size[1], fbank_size[2]

    # Step 3 : Time masking
    for i in range(time_mask_num):
        t = tf.random.uniform([], minval=0, maxval=
           time_masking_para, dtype=tf.int32)
        t0 = tf.random.uniform([], minval=0, maxval=tau-t,
           dtype=tf.int32)

        # mel_spectrogram[:, t0:t0 + t] = 0
        mask = tf.concat((tf.ones(shape=(1, n-t0-t, v, 1)),
                        tf.zeros(shape=(1, t, v, 1)),
                        tf.ones(shape=(1, t0, v, 1)),
                        ), 1)
        spectrogram = spectrogram * mask
    return tf.cast(spectrogram, dtype=tf.float32)
```

```
def spec_augment(spectrogram):

    v = spectrogram.shape[0]
    tau = spectrogram.shape[1]

    #warped_spectrogram = sparse_warp(spectrogram)

    warped_frequency_spectrogram = frequency_masking(
       spectrogram, v=v)

    warped_frequency_time_spectrogram = time_masking(
       warped_frequency_spectrogram, tau=tau)

    return warped_frequency_time_spectrogram

def visualization_tensor_spectrogram(spectrogram):
    """visualizing first one result of SpecAugment

    # Arguments:
      mel_spectrogram(ndarray): mel_spectrogram to visualize.
      title(String): plot figure's title
    """

    # Show mel-spectrogram using librosa's specshow.
    #plt.figure(figsize=(7, 5))
    fig = plt.figure(figsize=(7, 5), dpi=1000, frameon=False)
    ax = fig.add_axes([0,0,1,1], frameon=False)
    ax.axis('off')
    librosa.display.specshow(librosa.amplitude_to_db(
       spectrogram[0, :, :, 0], ref=np.max), y_axis='hz', fmax
       =8000, x_axis='time')
    #plt.colorbar(format='%+2.0f dB')
    #plt.title(title)
    #plt.tight_layout()
    plt.show()

# GENERATING REGULAR SPECTROGRAMS
audio_fpath = "./train_sounds/"
audio_clips = os.listdir(audio_fpath)
FIG_SIZE = (7, 5)

def generate_spectrogram():

    for i in audio_clips:
        spectrograms_path = "./aug_train_spects/"
```

```
        # to create augmented data "aug" was added before
            filename
        save_name = spectrograms_path + "aug" + i + ".jpg" #
            for aug data add "aug"

        if not os.path.exists(save_name):
            # check if a file already exists
            signal, sample_rate = librosa.load(audio_fpath + i,
                sr=2000)

            hop_length = 128 # in num. of samples
            n_fft = 2048 # window in num. of samples
            stft = librosa.stft(signal, n_fft=n_fft, hop_length
                =hop_length)

            # calculate abs values on complex numbers to get
                magnitude
            spectrogram = np.abs(stft)

            apply = SpecAugment(spectrogram, 'SM')
            time_warped = apply.time_warp()

            # Show time warped & masked spectrogram
            visualization_tensor_spectrogram(spectrogram=
                spec_augment(time_warped))

            plt.savefig(save_name, pil_kwargs={'quality': 95},
                bbox_inches=0, pad_inches=0)
            plt.close()
            librosa.cache.clear()

# Creating sprectrograms for both train and test batch
# for i in audio_clips:
#     spectrograms_path = "./aug_train_spects/"
#     save_name = spectrograms_path + i + ".jpg"
#     # check if a file already exists
#     if not os.path.exists(save_name):
#         signal, sample_rate = librosa.load(audio_fpath + i,
   sr=2000)
#         generate_spectrogram(signal, sample_rate, save_name)
#         plt.close()
generate_spectrogram()
```

B.4 augment.py

```
import random
import numpy as np
import tensorflow as tf
from tensorflow_addons.image import sparse_image_warp


class SpecAugment():
    '''
    Augmentation Parameters for policies
    -----------------------------------------
    Policy | W   | F   | m_F |   T   |   p   | m_T
    -----------------------------------------
    None   |   0 |   0 |   -   |   0   |   -   |   -
    -----------------------------------------
    LB     | 80 | 27 |   1   | 100 | 1.0 | 1
    -----------------------------------------
    LD     | 80 | 27 |   2   | 100 | 1.0 | 2
    -----------------------------------------
    SM     | 40 | 15 |   2   |  70 | 0.2 | 2
    -----------------------------------------
    SS     | 40 | 27 |   2   |  70 | 0.2 | 2
    -----------------------------------------

    LB  : LibriSpeech basic
    LD  : LibriSpeech double
    SM  : Switchboard mild
    SS  : Switchboard strong
    W   : Time Warp parameter
    F   : Frequency Mask parameter
    m_F : Number of Frequency masks
    T   : Time Mask parameter
    p   : Parameter for calculating upper bound for time mask
    m_T : Number of time masks
    '''

    def __init__(self, spectrogram, policy,
       zero_mean_normalized=True):
        self.spectrogram = spectrogram
        self.policy = policy
        self.zero_mean_normalized = zero_mean_normalized

        # Policy Specific Parameters
        if self.policy == 'LB':
            self.W, self.F, self.m_F, self.T, self.p, self.m_T
```

```
            = 80, 27, 1, 100, 1.0, 1
        elif self.policy == 'LD':
            self.W, self.F, self.m_F, self.T, self.p, self.m_T
            = 80, 27, 2, 100, 1.0, 2
        elif self.policy == 'SM':
            self.W, self.F, self.m_F, self.T, self.p, self.m_T
            = 8, 7, 1, 5, 0.2, 1
        elif self.policy == 'SS':
            self.W, self.F, self.m_F, self.T, self.p, self.m_T
            = 40, 27, 2, 70, 0.2, 2


    def time_warp(self):

        # Reshape to [Batch_size, time, freq, 1] for
            sparse_image_warp func.
        self.spectrogram = np.reshape(self.spectrogram, (-1,
            self.spectrogram.shape[0], self.spectrogram.shape
            [1], 1))

        v, tau = self.spectrogram.shape[1], self.spectrogram.
            shape[2]

        horiz_line_thru_ctr = self.spectrogram[0][v // 2]

        random_pt = horiz_line_thru_ctr[random.randrange(self.W
            , tau - self.W)] # random point along the horizontal
            /time axis
        w = np.random.uniform((-self.W), self.W) # distance

        # Source Points
        src_points = [[[v//2, random_pt[0]]]]

        # Destination Points
        dest_points = [[[v//2, random_pt[0] + w]]]

        self.spectrogram, _ = sparse_image_warp(self.
            spectrogram, src_points, dest_points,
            num_boundary_points=2)

        return self.spectrogram


    def freq_mask(self):

        v = self.spectrogram.shape[1] # no. of mel bins
```

```
        # apply m_F frequency masks to the mel spectrogram
        for i in range(self.m_F):
            f = int(np.random.uniform(0, self.F)) # [0, F)
            f0 = random.randint(0, v - f) # [0, v - f)
            self.spectrogram[:, f0:f0 + f, :, :] = 0

        return self.spectrogram


    def time_mask(self):

        tau = self.spectrogram.shape[2] # time frames

        # apply m_T time masks to the mel spectrogram
        for i in range(self.m_T):
            t = int(np.random.uniform(0, self.T)) # [0, T)
            t0 = random.randint(0, tau - t) # [0, tau - t)
            self.spectrogram[:, :, t0:t0 + t, :] = 0

        return self.spectrogram
```

B.5 class_separation.py

```
import os
os.environ['LIBROSA_CACHE_DIR'] = '/tmp/'
import pandas as pd
import shutil
import sys

# READING THE LABELS
# def append_ext(fn):
#     return fn + ".jpg"

traindf=pd.read_csv("./train.csv", dtype=str)

# SEPARATE IMAGES INTO CLASSES
train_dir = "./train_spectrograms"
# test_dir = "./test_2022"
# creating separating directory
classes = "./train/"
# classes = "./test/"

# if the folder does not exist create it
if not os.path.exists(classes):
    os.mkdir(classes)
```

```
for filename, class_name in traindf.values:
    # Create subdirectory with ‘class_name‘
    if not os.path.exists(classes + str(class_name)):
        os.mkdir(classes + str(class_name))
    src_path = train_dir + '/'+ filename + '.jpg'
    dst_path = classes + str(class_name) + '/' + filename + '.
       jpg'
    try:
        shutil.copy(src_path, dst_path)
        print("Sucessful")
    except IOError as e:
        print('Unable to copy file {} to {}'
              .format(src_path, dst_path))
    except:
        print('When try copy file {} to {}, unexpected error:
           {}'
              .format(src_path, dst_path, sys.exc_info()))
```

B.6 dataset_split.py

```
import os
os.environ['LIBROSA_CACHE_DIR'] = '/tmp/'
import splitfolders

# SPLIT TRAIN FOLDER TO TRAIN AND VALIDATION
splitfolders.ratio("train", output="train_val_split",
    seed=1337, ratio=(.8, .2), group_prefix=None, move=True)
```

B.7 mobilenetv2_model.py

```
import matplotlib.pyplot as plt
import os

os.environ['LIBROSA_CACHE_DIR'] = '/tmp/'
import pandas as pd
import tensorflow as tf
from tensorflow import keras

traindf = pd.read_csv("./train.csv", dtype=str)
# traindf["clip_name"]=traindf["clip_name"].apply(append_ext)
# traindf.head()
```

```
# LOADING IMAGES
train_path = "./train_val_split/train/"
validation_path = "./train_val_split/val/"
# test_path = "./test_2022/"

train_dir = os.path.join(train_path)
validation_dir = os.path.join(validation_path)
# test_dir = os.path.join(test_path)

BATCH_SIZE = 16
IMG_SIZE = (160, 160)

train_dataset =
tf.keras.utils.image_dataset_from_directory(train_dir,
                                            shuffle=True,
                                            batch_size=
                                               BATCH_SIZE,
                                            image_size=IMG_SIZE
                                               )

validation_dataset = tf.keras.utils.
   image_dataset_from_directory(validation_dir,
                                            shuffle=True,
                                            batch_size=
                                               BATCH_SIZE,
                                            image_size=IMG_SIZE
                                               )

# CREATING A TEST SET FROM THE VALIDATION SET
val_batches = tf.data.experimental.cardinality(
   validation_dataset)
test_dataset = validation_dataset.take(val_batches // 5)
validation_dataset = validation_dataset.skip(val_batches // 5)

# BUFFERED PREFETCHING TO LOAD IMGS FROM DISK
AUTOTUNE = tf.data.AUTOTUNE

train_dataset = train_dataset.prefetch(buffer_size=AUTOTUNE)
validation_dataset = validation_dataset.prefetch(buffer_size=
   AUTOTUNE)
test_dataset = test_dataset.prefetch(buffer_size=AUTOTUNE)

# DATA AUGMENTATION FOR MFCC
# data_augmentation = tf.keras.Sequential([
#     # tf.keras.layers.RandomFlip('horizontal'),
#     # tf.keras.layers.RandomRotation(0.2),
#     tf.keras.layers.RandomWidth(factor=(0.2, 0.3),
```

```
                                    interpolation='gaussian')
# ])

# USING PREPROCESSING TO RESCALE PIXEL VALUES
preprocess_input = tf.keras.applications.mobilenet_v2.
    preprocess_input
rescale = tf.keras.layers.Rescaling(1. / 127.5, offset=-1)

# Create the base model from the pre-trained model MobileNet V2
IMG_SHAPE = IMG_SIZE + (3,)
base_model = tf.keras.applications.MobileNetV2(input_shape=
    IMG_SHAPE,
                                                include_top=
                                                    False,
                                                weights='
                                                    imagenet')

# FEATURE EXTRACTOR CONVERTS EACH IMAGE INTO A 5*5*1280 BLOCK
    OF FEATURES
image_batch, label_batch = next(iter(train_dataset))
feature_batch = base_model(image_batch)
print(feature_batch.shape)

# freezing the convolutional base
base_model.trainable = False

# converting the features to a single 1280-element vector per
    image
global_average_layer = tf.keras.layers.GlobalAveragePooling2D()
feature_batch_average = global_average_layer(feature_batch)
print(feature_batch_average.shape)

# converting features into a single prediction per image
prediction_layer = tf.keras.layers.Dense(1)
prediction_batch = prediction_layer(feature_batch_average)
print(prediction_batch.shape)

# BUILDING THE MODEL
inputs = tf.keras.Input(shape=(160, 160, 3))
# x = data_augmentation(inputs)
# x = preprocess_input(x)
x = preprocess_input(inputs)
x = base_model(x, training=False)
x = global_average_layer(x)
x = tf.keras.layers.Dropout(0.2)(x)
outputs = prediction_layer(x)
model = tf.keras.Model(inputs, outputs)
```

```
# compiling the model
base_learning_rate = 0.0001
model.compile(optimizer=
            tf.keras.optimizers.Adam(learning_rate=
                base_learning_rate),
            loss=tf.keras.losses.BinaryCrossentropy(from_logits
                =True),
            metrics=['accuracy'])

# TRAINING THE BASE MODEL
initial_epochs = 20

history = model.fit(train_dataset,
                    epochs=initial_epochs,
                    validation_data=validation_dataset)

# PLOTTING THE RESULTS
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']

loss = history.history['loss']
val_loss = history.history['val_loss']

plt.figure(figsize=(8, 8))
plt.subplot(2, 1, 1)
plt.plot(acc, label='Training Accuracy')
plt.plot(val_acc, label='Validation Accuracy')
plt.legend(loc='lower right')
plt.ylabel('Accuracy')
plt.ylim([min(plt.ylim()), 1])
plt.title('Training and Validation Accuracy')

plt.subplot(2, 1, 2)
plt.plot(loss, label='Training Loss')
plt.plot(val_loss, label='Validation Loss')
plt.legend(loc='upper right')
plt.ylabel('Cross Entropy')
plt.ylim([0, 1.0])
plt.title('Training and Validation Loss')
plt.xlabel('epoch')
plt.show()

fn = "9th_training_20epochs_freq_mask"
plt.savefig(fn, format="png")
print(f"Saving '{fn}.png'")
```

```
# EVALUATING THE MODEL
loss1, accuracy1 = model.evaluate(validation_dataset)

print("Validation loss: {:.2f}".format(loss1))
print("Validation accuracy: {:.2f}".format(accuracy1))

# unfreezing the convolutional base
base_model.trainable = True

# fine tuning from the 100th layer
fine_tune_at = 100

# Freeze all the layers before the `fine_tune_at` layer
for layer in base_model.layers[:fine_tune_at]:
                    layer.trainable = False

# setting lower learning rate to reduce overfitting
model.compile(loss=tf.keras.losses.BinaryCrossentropy(
   from_logits=True),
              optimizer=tf.keras.optimizers.RMSprop(
                                learning_rate=
                                   base_learning_rate/10),
                                metrics=['accuracy'])

fine_tune_epochs = 10
total_epochs = initial_epochs + fine_tune_epochs

history_fine = model.fit(train_dataset,
                         epochs=total_epochs,
                         initial_epoch=history.epoch[-1],
                         validation_data=validation_dataset)

acc += history_fine.history['accuracy']
val_acc += history_fine.history['val_accuracy']

loss += history_fine.history['loss']
val_loss += history_fine.history['val_loss']

plt.figure(figsize=(8, 8))
plt.subplot(2, 1, 1)
plt.plot(acc, label='Training Accuracy')
plt.plot(val_acc, label='Validation Accuracy')
plt.ylim([0.6, 1])
plt.plot([initial_epochs-1,initial_epochs-1],
          plt.ylim(), label='Start Fine Tuning')
plt.legend(loc='upper left')
plt.title('Training and Validation Accuracy')
```

```
plt.subplot(2, 1, 2)
plt.plot(loss, label='Training Loss')
plt.plot(val_loss, label='Validation Loss')
plt.ylim([0, 1.0])
plt.plot([initial_epochs-1,initial_epochs-1],
          plt.ylim(), label='Start Fine Tuning')
plt.legend(loc='upper left')
plt.title('Training and Validation Loss')
plt.xlabel('epoch')
plt.show()

fn = "9th_training_20epochs_fine_tuned_dropout20_freq_mask"
plt.savefig(fn, format="png")
print(f"Saving '{fn}.png'")

# EVALUATING THE MODEL
loss, accuracy = model.evaluate(test_dataset)
print('Test loss:', loss)
print('Test accuracy:', accuracy)

# RETRIEVE IMAGES FROM THE TEST SET
image_batch, label_batch = test_dataset.as_numpy_iterator().
   next()
 predictions = model.predict_on_batch(image_batch).flatten()

# Apply a sigmoid since the model returns logits
predictions = tf.nn.sigmoid(predictions)
predictions = tf.where(predictions < 0.5, 0, 1)

print('Predictions:\n', predictions.numpy())
print('Labels:\n', label_batch)
```

Appendix C

Appendix C - Training results

Figure C.1: 10 epochs only fine tuning of the model - *BATCH_SIZE*=32, accuracy was 89%

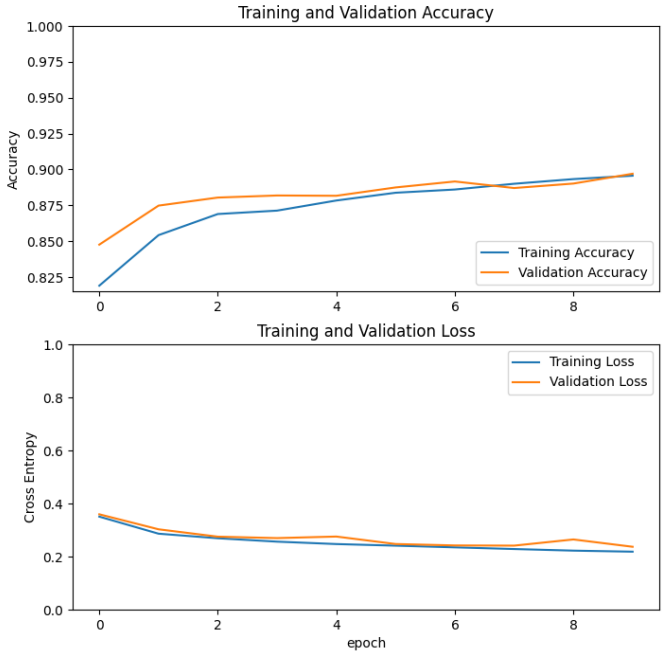


Figure C.2: 10 epochs + 10 epochs with fine tuning - *BATCH_SIZE*=32, accuracy was 87%

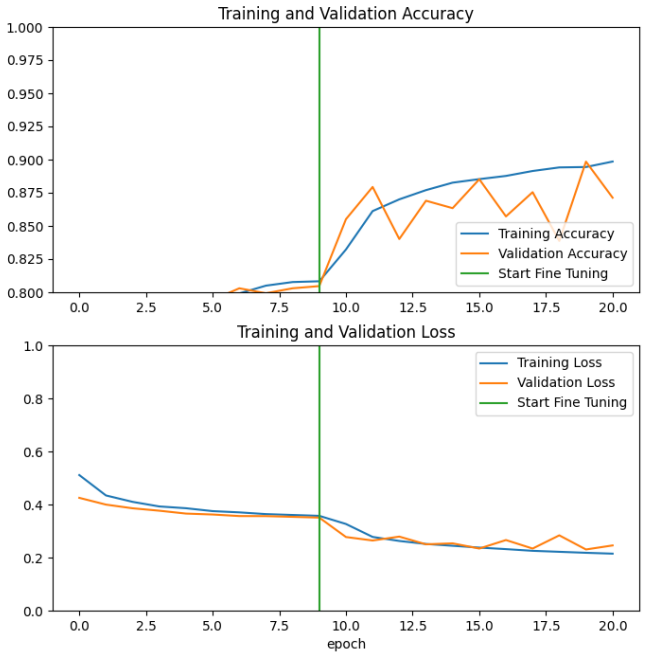


Figure C.3: 10 epochs + 10 epochs with fine tuning - *BATCH_SIZE*=100, accuracy was 88%

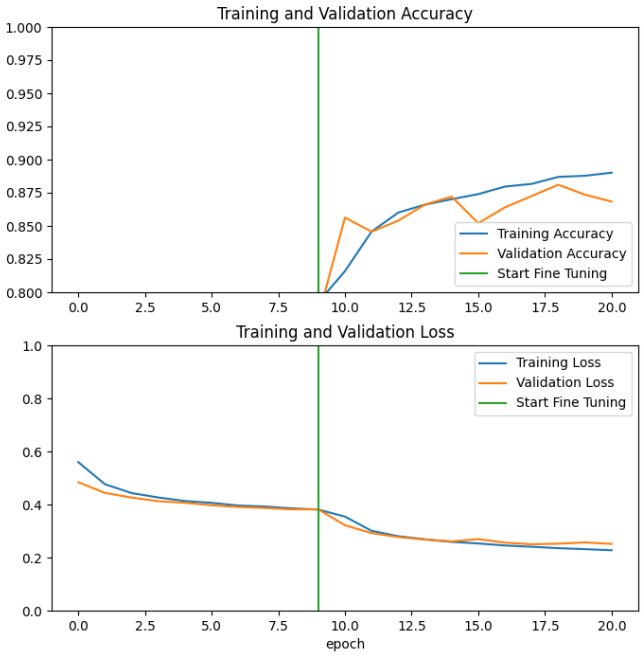


Figure C.4: 10 epochs + 10 epochs with fine tuning - *BATCH_SIZE*=32, layers.Dropout(0.4), accuracy was 86%

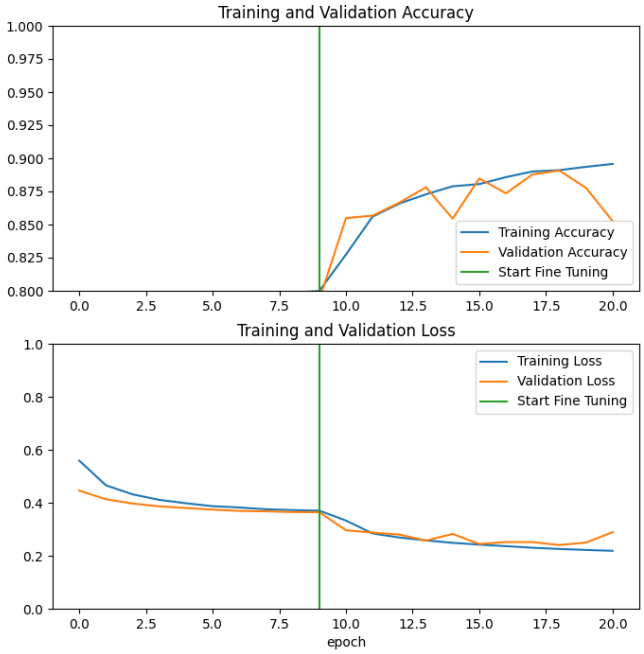


Figure C.5: 10 epochs + 10 epochs with fine tuning and data augmentation - *BATCH_SIZE*=16, accuracy was 88%

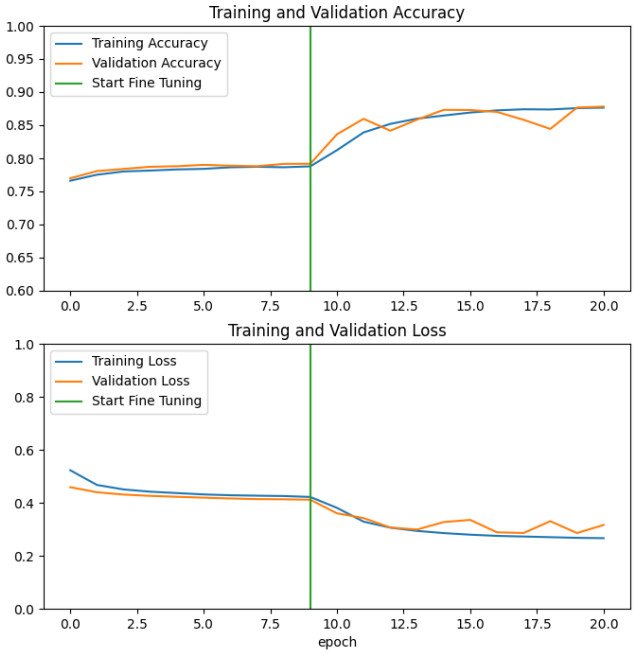


Figure C.6: 20 epochs + 20 epochs with fine tuning and pre-augmented data - *BATCH_SIZE*=32, layers.Dropout(0.4), accuracy was 86%