Highly Efficient Sound Classification for Marine Mammals

by

Xiangrui Liu

B.Sc., The University of British Columbia, 2020

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE

in
THE COLLEGE OF GRADUATE STUDIES
(Electrical Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA
(Okanagan)

January 2024

© Xiangrui Liu 2024
The following individuals certify that they have read, and recommend to the College of Graduate Studies for acceptance, the thesis entitled:
Highly Efficient Sound Classification for Marine Mammals submitted by
Xiangrui Liu in partial fulfillment of the requirements for
the degree of Master of Applied Science.

Dr. Julian Cheng, School of Engineering
Supervisor

Dr. Shan Du, Irving K Barber Faculty of Science
Co-supervisor

Dr. Jahangir Hossain, School of Engineering
Supervisory Committee Member

Dr. Thomas Johnson, School of Engineering
Supervisory Committee Member

Dr. Mahmudur Fatmi, School of Engineering
External Examiner
Abstract

Marine mammals and their ecosystem face significant threats from, for example, military active sonar and marine transportation. To mitigate this harm, early detection and classification of marine mammals are essential. Recent solutions involve spectrogram comparison and machine learning. However, the solutions show weaknesses in efficiency. Therefore, we propose a novel knowledge distillation framework, named XCFSMN, for this problem. We construct a teacher model that fuses the features extracted from an X-vector extractor, a DenseNet, and a Cross-Covariance attended compact Feed-Forward Sequential Memory Network (cFSMN). The teacher model transfers knowledge to a simpler cFSMN model through a temperature-cooling strategy for efficient learning. Compared to multiple convolutional neural network backbones and transformers, the proposed framework achieves state-of-the-art efficiency and performance. The improved model size is approximately 25 times smaller and the inference time is 27 times shorter on average without affecting the model’s accuracy significantly.
Lay Summary

Marine mammals, like dolphins and whales, are facing serious threats from things like loud military sonar and maritime traffic. To help protect them, it is crucial to rapidly detect and identify these animals. Traditional approaches involve human inspection using a passive acoustic monitor or telescope. More recent solutions use spectrogram comparison and automatic classification through machine learning. However, the existing solutions are inefficient as they may require the presence of specialists; they are susceptible to weather conditions; and they tend to be slow in response.

In alignment with the advancements in machine learning, we propose a deep-learning framework named XCFSMN to address the problem efficiently and accurately. The framework consists of two neural networks: a powerful yet computationally expensive teacher model and a smaller, faster student model. The teacher model is complex and powerful but at the expense of inference speed and size. In other words, it takes time to identify the sound and it may be too big to deploy onto most vessels. Therefore, we construct a smaller student model that learns from the power teacher model. Through such a knowledge transfer process, the student model is able to attain 90% of the knowledge from the teacher model and makes a prediction at a speed that may be 27 times faster than the teacher model on average. In summary, the framework achieves state-of-the-art efficiency and performance, marking a step forward in protecting these animals and their homes in the ocean.
Preface

A list of my conference publications related to this thesis at The University of British Columbia is provided in the following.


1 Xiangrui Liu and Julian Cheng, A Highly Efficient Marine Mammals Classifier Based on a Cross-Covariance Attended Compact Feed-Forward Sequential Memory Network (Student Abstract), in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 16268–16269
# Table of Contents

Absekt .................................................. iii
Lay Summary .......................................... iv
Preface .................................................. v
Table of Contents ....................................... vi
List of Tables .......................................... viii
List of Figures ......................................... ix
Acknowledgements ...................................... xi

Chapter 1: Introduction .................................. 1

Chapter 2: Background ................................... 5
  2.1 Threats to Cetacean Populations ....................... 5
  2.2 Related Work ......................................... 7
    2.2.1 Acoustic Signal Processing and Representation .... 7
    2.2.2 Machine Learning Based Methods ....................... 8
    2.2.3 Deep Learning Based Methods ......................... 8
    2.2.4 Knowledge Distillation ............................... 12
  2.3 Comparison Models .................................... 13
    2.3.1 AlexNet ............................................ 13
    2.3.2 VGG-16 ............................................ 14
    2.3.3 ResNet ............................................ 16
TABLE OF CONTENTS

2.3.4 Data-efficient Image Transformers (DeiT) . . . . . . . 17
2.3.5 Longformer . . . . . . . . . . . . . . . . . . . . . . . . 22
2.3.6 FNet . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24

Chapter 3: Methodology . . . . . . . . . . . . . . . . . . . . . 25
3.1 Overall Structure of The Framework . . . . . . . . . . . . . 25
3.2 Feature Extraction and Pre-processing . . . . . . . . . . . . 26
  3.2.1 FBank Extraction . . . . . . . . . . . . . . . . . . . . 26
  3.2.2 Cepstral Mean and Variance Normalization (CMVN) . 26
3.3 Teacher Model . . . . . . . . . . . . . . . . . . . . . . . . 28
  3.3.1 DenseNet-121 . . . . . . . . . . . . . . . . . . . . . . . 28
  3.3.2 TDNN . . . . . . . . . . . . . . . . . . . . . . . . . . . 31
  3.3.3 Modified Student Model Unit . . . . . . . . . . . . . . 32
3.4 Student Model (CC-FSMN) . . . . . . . . . . . . . . . . . . . 33
  3.4.1 Compact Feed-Forward Sequential Memory Network . 33
  3.4.2 Cross-Covariance Attention (XCA) . . . . . . . . . . . 35
3.5 Loss Function . . . . . . . . . . . . . . . . . . . . . . . . . 37

Chapter 4: Experimental Setup . . . . . . . . . . . . . . . . . . 39
4.1 Dataset . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 39
4.2 Experiment Details . . . . . . . . . . . . . . . . . . . . . . . 39
4.3 Evaluation Metrics . . . . . . . . . . . . . . . . . . . . . . . 41

Chapter 5: Results Comparison . . . . . . . . . . . . . . . . . . 43
  5.1 Experiment Results . . . . . . . . . . . . . . . . . . . . . . 43
  5.2 Ablation Study . . . . . . . . . . . . . . . . . . . . . . . . 46

Chapter 6: Conclusion and Future Work . . . . . . . . . . . . . 51

Bibliography . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 52

Appendix . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 59
  Appendix A: Tables . . . . . . . . . . . . . . . . . . . . . . . 59
List of Tables

Table 2.1 The trend of whale population from 1890 to 2001 . . . 5
Table 4.1 Audio length of the five classes in hours. . . . . . . . 40
Table 5.1 Comparative results of XCFSMN and the baselines on efficiency. ‘TT’ denotes the training time in minutes and ‘IT’ denotes the inference time in seconds. . . . . . . . . 43
Table 5.2 Comparative results of 12-class WMMSD. . . . . . . . 44
Table 5.3 Comparative results of 5-class WMMSD. . . . . . . . 44
Table 5.4 Comparative results of 5-class Moby. . . . . . . . . . 45
Table 5.5 Comparative results of 5-class merged. . . . . . . . . 45
Table 5.6 Results of the ablation study on the student model. ‘TT’ denotes the training time in minutes and ‘IT’ denotes the inference time in seconds. . . . . . . . . . 48
Table 5.7 Results of the ablation study on the KD temperature strategy. ‘TT’ denotes the training time in minutes and ‘IT’ denotes the inference time in seconds. . . . . . . . . . 49
Table 6.1 TOPSIS score of 5-class Moby data. . . . . . . . . . . 59
Table 6.2 TOPSIS score of 5-class Watkins data. . . . . . . . . . 60
Table 6.3 TOPSIS score of 5-class merged data. . . . . . . . . . . 60
Table 6.4 TOPSIS score of 12-class Watkins data. . . . . . . . . 61
Table 6.5 Overall TOPSIS score. . . . . . . . . . . . . . . . . . . 61
# List of Figures

| Figure 2.1 | Sample convolution of a 7x7 pixel RGB image. | 9 |
| Figure 2.2 | The architecture of a standard transformer. Adapted from [51]. | 11 |
| Figure 2.3 | AlexNet architecture. | 14 |
| Figure 2.4 | VGG architecture. | 15 |
| Figure 2.5 | The residual block. | 17 |
| Figure 2.6 | The ResNet-50 architecture. Adapted from [18]. | 18 |
| Figure 2.7 | Deit structure. | 21 |
| Figure 2.8 | Architecture of a Longformer. | 22 |
| Figure 2.9 | Demonstration of different attention maps. | 23 |
| Figure 2.10 | Structure of FNet. | 24 |
| Figure 3.1 | The overall flow of XCFSMN. | 25 |
| Figure 3.2 | The flow of FBanks extraction. | 27 |
| Figure 3.3 | The structure of the teacher model. | 29 |
| Figure 3.4 | FBank spectrogram of bowhead whale and minke whale. | 29 |
| Figure 3.5 | The structure of a DenseNet. | 31 |
| Figure 3.6 | The structure of the teacher model. | 32 |
| Figure 3.7 | The structure of the student model. | 33 |
| Figure 3.8 | The structures of FSMN (left) and cFSMN (right) networks. | 34 |
| Figure 3.9 | The illustration of estimating attention in XCA. | 35 |
| Figure 5.1 | Confusion matrix for student model’s results on combined dataset. | 47 |
| Figure 5.2 | Spectrograms of three whale species. | 48 |
LIST OF FIGURES

Figure 5.3 Visualization of features using a t-SNE plot for 12-class Watkins experiment. . . . . . . . . . . . . . . . . . . . . . . 50
Acknowledgements

I would like to thank my advisor, Professor Julian Cheng, and Professor Shan Du for their invaluable mentorship, guidance, and support. I would also like to thank my family for their support and encouragement.
Chapter 1

Introduction

Marine mammals play an important role in marine ecosystems [6]. Marine mammals contribute to the overall health and stability of the ecosystem in the following ways. The top predators, such as killer whales and certain dolphins play essential roles in regulating their prey’s population. Further, marine mammals can also be indicators of the ecosystem’s health. As predators in the upper level of the food webs, they accumulate some of the highest contaminants [11]. The accumulated contaminants and pollutants in their bodies can be studied by scientists to identify potential threats in marine ecosystems.

However, marine mammals’ population has been declining sharply [48]. Besides the loss of habitats due to man-made construction, over-fishing, and human activities, such as active sonars and vessels, also cause the marine mammal population to decrease [8]. The sound waves generated by military sonar can severely injure and even kill marine mammals [39]. Moreover, the active sonar frequency interferes with the echolocation used by cetaceans for localization and hunting. Thus, the protection of marine mammals is much needed.

One direction to protect marine mammals is to achieve advanced detection and classification so vessels can make a detour, and the navy can reschedule their exercises to avoid harming the protected marine mammals. Machine learning has been introduced to detect and classify marine mammals and it can be used to tackle the problem in two ways. One approach is to analyze the photos of marine mammals using machine learning models. For example, the Simple Linear Iterative Clustering technique was used to segment the aerial images of dugongs [32] and then train these images with convolutional neural networks (CNNs), obtaining an F1 score of 0.406.
Besides aerial images, an attempt has been made to compare three machine-learning methods using Commerson’s dolphins’ photos [41], and the F1 score of the best model is 0.833. Another approach classifies marine mammals using the vocalizations captured by passive acoustic monitors (PAM) since marine mammals produce sounds of different frequencies and patterns. With the advantages of the complex networks and weights trained using large datasets, various CNNs [1, 12, 30, 58] have been used to build marine mammals’ sound classifiers to produce accurate predictions.

Indeed, machine learning has attracted interest as it can identify marine mammals’ sounds accurately without the presence of specialists. There is still room for improvement. First, existing works mainly focus on applying transfer learning using different pre-trained models. These models are commonly made up of complex and deep neural networks with the advantage of being accurate but at the expense of efficiency. In other words, their training time and inference time can be shortened. Second, most existing marine mammals’ sound classification work using deep learning is trained using private datasets, so the signal-noise ratio (SNR) and data size are unknown. The only known open-sourced dataset used was the Watkins Marine Mammals Sound Database (WMMSD)\(^1\). The experiment that used WMMSD [30] only tested the model on the best cuts of the database, which are the audios with high SNR. The noisy audio cuts were not explored; hence the robustness of the models is unknown. Therefore, our goal is to develop a computationally friendly and robust framework that recognizes marine mammals’ acoustic signals efficiently.

This thesis proposes a knowledge distillation (KD) marine mammals’ sound classification framework that is much more efficient than some well-known CNN-based models and transformers. We introduce a Cross-Covariance attended compact Feed-Forward Sequential Memory Network (CC-FSMN) that can efficiently learn the acoustic features. On the basis of the CC-FSMN, we build a teacher model that embeds an X-vector extractor and a DenseNet network to better learn the features and increase the accuracy of the system. We further replace the Dropout mechanism in the Cross-

\(^1\)https://cis.whoi.edu/science/B/whalesounds/index.cfm
Covariance Attention (XCA) layer with a DropKey mechanism to improve the robustness of the model. The relatively large and complex teacher network effectively improves the CC-FSMN-based student model’s accuracy through an effective knowledge transfer. Compared to a pure CC-FSMN, the student model has improved robustness, F1 score, and an identical inference time. To verify the generalizability and performance of the system, two datasets are included: WMMSD and the MobySound Database [34]. We compare the efficiency and accuracy of multiple popular models, including AlexNet [30], VGG-16 [58], DenseNet [58], ResNets [1, 18], DeiT III [50], FNet [27], and Longformer [54].

We can summarize our contributions as follows:

− We propose a Cross-Covariance Attended Compact Feed-Forward Sequential Memory Network (CC-FSMN) that is excellent in handling acoustic signal classification. It is the base architecture for the teacher and student model.

− We propose a teacher model that fuses features extracted from DenseNet-121, TDNN, and CC-FSMN. Through extensive empirical evaluations and comparisons, the teacher achieves state-of-the-art performance.

− We apply a temperature cooling strategy to the knowledge distillation approach to mitigate the knowledge gap caused by model size and architecture differences between the teacher and the student model, increasing the effectiveness of the process.

− Through knowledge distillation, the student model, which is approximately 25 times smaller than the teacher model, is well-trained. The student model has the best inference speed while retaining a very close accuracy to the teacher model. The student model’s highly compressed size makes its deployment on various vessels applicable.

The structure of the whole thesis is divided into five chapters. Chapter 1 provides an introduction to the thesis, while Chapter 2 details the background information, including the threats posed to marine mammals,
existing solutions to these threats, and an introduction of the comparison models. Chapter 3 discusses the structure of the proposed framework in detail, including the feature extraction process, the teacher model architecture, the student model architecture, and the loss functions. Chapter 4 presents the specific experimental setup for the experiments, details of the datasets, and evaluation metrics used. Chapter 5 presents and discusses the results of the models and an ablation study of the proposed framework. Finally, Chapter 6 concludes the thesis and recommends potential future work based on the results and conclusion from the thesis work.
Chapter 2

Background

2.1 Threats to Cetacean Populations

It is challenging to assess the precise number of whale populations. The estimation of some whale species shown in Table 2.1 [2] suggests the overall populations of the cetaceans have dropped sharply. Shockingly, certain species have their populations reduced by more than 80%. Adding to these concerns, at least 25% of the world’s cetaceans were recently confirmed as endangered and the situation is not optimistic as the status of many others remains unknown.

Table 2.1: The trend of whale population from 1890 to 2001

<table>
<thead>
<tr>
<th>Species</th>
<th>1890</th>
<th>2001</th>
<th>Relative Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blue Whale</td>
<td>340,280</td>
<td>4,727</td>
<td>−99%</td>
</tr>
<tr>
<td>Bowhead Whale</td>
<td>89,000</td>
<td>9,450</td>
<td>−89%</td>
</tr>
<tr>
<td>Right Whale</td>
<td>84,100</td>
<td>9,239</td>
<td>−89%</td>
</tr>
<tr>
<td>Fin Whale</td>
<td>762,400</td>
<td>109,600</td>
<td>−86%</td>
</tr>
<tr>
<td>Humpback Whale</td>
<td>231,700</td>
<td>42,070</td>
<td>−82%</td>
</tr>
<tr>
<td>Sei/Bryde’s Whale</td>
<td>392,300</td>
<td>181,490</td>
<td>−54%</td>
</tr>
<tr>
<td>Gray Whale</td>
<td>24,600</td>
<td>15,936</td>
<td>−35%</td>
</tr>
<tr>
<td>Minke Whale</td>
<td>637,000</td>
<td>506,900</td>
<td>−20%</td>
</tr>
</tbody>
</table>

The decrease in cetacean populations can be traced back centuries. In the past, people hunted them for food, warmth, and luxuries such as perfume [2]. The advent of steam-allowed boats marked the beginning of the modern whaling era. The faster and more efficient fishing vessels led to the expansion of species and numbers being caught. Other than commercial whaling, climate change is also a significant factor that has decreased the cetacean
2.1. THREATS TO CETACEAN POPULATIONS

populations. Temperature is a key factor that determines the cetaceans' distribution as the species evolved to live in areas of a certain temperature range. Some whale species exhibit a degree of adaptability to shifting climates but most of them remain incapable of doing that. For instance, the endangered vaquitas’ habitat is limited to the warm waters located at the embayment of California. Such an enclosed landscape restricts the vaquitas' movement to find cooler water when climate changes [43]. There are other indirect impacts of climate change, for example, alterations in availability, locality, and abundance of food. This phenomenon particularly imperils species reliant on specialized feeding habitats. For instance, in the North Atlantic, warm water species population is rising due to climate warming and the effects of the North Atlantic Oscillation [15]. This is detrimental as the warm water species are not replacing the cold water species in similar abundance. It is also suggested that there may also be a correlation between climate-induced reductions in prey and numbers of starving harbor porpoises stranding on the Scottish North Sea coast [22, 26, 31].

The deployment of military sonar also poses a significant and pervasive threat to cetacean populations. There were at least 16 whales of 3 species stranded in the northern Bahamas when the US Navy vessels passed by that area [3]. The Bahamas incident in 2000 was not the only stranding caused by the navy activities. There are more stranding of whales caused by the navies. Across the Pacific, 11 mass strandings on the central Pacific coast of Honshu in Japan were reported and the strandings happened in the area adjacent to a US Navy command base [7]. The Scientific Committee of the International Whaling Commission (IWC) suggests that “these whales died from acoustic or blast trauma that may have been caused by exposure to naval activities south of Taiwan” [10]. In 2005, another 37 whales were stranded in the Outer Bank, North Carolina after a US Navy vessel used active sonar for about 7 minutes [24]. Similarly, more than a hundred long-finned pilot whales were stranded in Australia after the Australian Navy used high-frequency sonar [38]. It was later concluded that the military sonar deployed by the Navy caused these strandings.

Mass strandings represent just one facet of the adverse effects inflicted
2.2. RELATED WORK

upon cetaceans by active sonar. Other abnormal conditions include, but are
not limited to changes in singing behavior [36, 42], keeping silent [52], and
the appearance of deep-water animals in the shallow water [20, 23]. The
common thing to them is the presence of navy and military sonar. It is also
proven that besides the interference between the cetaceans’ sounds and the
active sonar, the high acoustic energy can directly and physically damage
the cetaceans’ auditory system and even result in fatal consequences.

2.2 Related Work

2.2.1 Acoustic Signal Processing and Representation

A common approach for identifying the sounds of marine mammals in-
volves studying the characteristics of the interested acoustic signals. One
method is to find the spectrogram correlation [33], calculating the similarity
values between the correlation kernels of the templates’ spectrograms and
the input signals’ spectrograms. Another method compares the raw data
with templates of the species. An example of the approach is the sound-
comparative method, which evaluates the template and the interested signal
using a preset threshold value [9].

Another approach involves extracting useful information and features
from the spectrograms. For example, a whale detection and classification
system that estimates the time variation of the fundamental frequency as a
pitch track [4]. Then the system extracts the attributes from the pitch track
for classification using quadratic discriminant function analysis. Besides
preprocessing of acoustic signals, representation learning can significantly
affect the performance of audio signal classification [53]. A new way to rep-
resent spectrograms of marine mammals’ acoustic signals has been proposed
[49]. The interpolated and stacked spectrograms are generated by running a
Short-time Fourier Transform with various parameters to evaluate the fea-
2.2. RELATED WORK

2.2.2 Machine Learning Based Methods

Several works [13, 16, 17, 35] have applied machine learning to tackle the marine mammal classification problem. For example, a self-organizing feature map was used to classify the vocalizations of Humpback whales [35]. Some approaches involve comparing methods such as a regression tree, a standard fully connected network, and a restricted Boltzmann [13, 17]. Instead of constructing a complex network, another way to increase the accuracy of a classifier is to enhance the input data. For instance, spectrogram noise removal through a median filter, average subtraction, and Gaussian smoothing kernel to enhance the recorded odontocete whistles and trained using a linear classifier [16].

2.2.3 Deep Learning Based Methods

With the advancement of deep learning, transfer learning based on CNNs is applied to classify marine mammals based on acoustic signals. The CNN’s capability of discriminating spectro-temporal information from spectrograms makes it an ideal network for processing acoustic information. It has been widely used in acoustic classification tasks such as bio-acoustics classification, environmental sound classification, and underwater sonar image classification [12]. The main body of a CNN is usually made up of Convolution layers, Pooling layers, and fully connected layers. To illustrate how CNN studies the pattern of an image, see Figure ?? . The image is digitalized to be a matrix of numbers to the computer and each pixel’s value is determined by the magnitude of the color at that pixel. The convolutional layers apply a set of filters that slide through the image matrix. At each position, the filter computes a weighted sum of the pixel values in its receptive field as shown in Figure 2.1. This process allows the model to learn the local patterns and features such as edges. It is common to have a Pooling layer after the convolutional layer to downsample the features reducing computational complexity. At the end of a series of convolutional layers and Pooling layers, there will be a few fully connected layers that behave as the conventional neural network that classifies the learned features.
Figure 2.1: Sample convolution of a 7x7 pixel RGB image.
2.2. RELATED WORK

Besides its promising performance on visual representation classification tasks, another reason that drives the research focus to shift to deep learning is that automatic detection and classification system based on deep learning outperforms human analysis in efficiency. For instance, different fine-tuned and pre-trained CNN backbones are used to build accurate marine mammals’ sound classifiers [1, 30, 58]. However, these works mainly focus on limited whale species and are relatively inefficient. A different and novel approach turns the classification task into a regression problem. Instead of a classifier, a regressor is built using YOLO to predict bounding boxes in the spectrograms [12].

Later, the appearance of self-attention mechanisms and transformers changed deep learning significantly. The first transformer stacks attention layers and surpasses the best result of that time [51]. The standard transformer shown in Figure 2.2 consists of two components, an encoder on the left and a decoder on the right. The encoder consists of $N$ identical blocks and each block is made up of an attention layer and a Fully Connected Feed-Forward Network with two linear transformations and ReLU activation:

$$\text{FFN}(x) = \text{ReLU}(W_1x + b_1)W_2 + b_2.$$  

(2.1)

The $N$ Transformer encoders apply the same linear transformations to the input sequence using different weight and bias parameters. The transformer architecture does not capture the position information solely and the introduction of positional encoding is required. The positional encoding vectors have the same dimension as the input embeddings and are generated using sine and cosine functions of different frequencies. The positional information is obtained by summing the positional vector and the input embeddings. The decoder has the same number of $N$ as the encoder, and it is made up of three sublayers. Different from the encoder which is designed to attend to all features in the input, the decoder only attends to preceding words if one is doing NLP tasks. The prediction for a word only depends on known outputs that come before it. On the decoder side, this attention layer receives the queries from the previous decoder sublayer and the keys and values from
the output of the encoder. Hence, the decoder can attend to all features in the input sequence. The decoder block ends with a Fully Connected Network like the encoder. Furthermore, the three sublayers also have residual connections around them and a normalization layer. In addition, positional encodings are also injected into the input embeddings of the decoder in the same manner as the encoder.

Figure 2.2: The architecture of a standard transformer. Adapted from [51].
2.2. RELATED WORK

The idea of transformers inspires researchers to develop more transformer architectures [21]. For instance, a newly proposed model, DeiT III [50], is a variant of ViT that includes a new data-augmentation procedure that involves Gaussian Blur, solarization, and grayscale, and it obtains a competitive performance in image classification. Indeed, CNN-based and transformer models produce some decent classifiers but they are computationally costly. To speed up the full attention transformer, Longformer [5] specifies the full self-attention matrix making it a more efficient architecture for long sequence data. However, the transformers require large amounts of data and strong computational power to be effective. FNet [27] is an attention-free transformer architecture that replaces attention layers with a Fourier mixing sublayer with a feed-forward sublayer to increase the efficiency of the transformer. However, the FNet transformer obtains improved efficiency at the expense of accuracy.

2.2.4 Knowledge Distillation

Big models show outstanding accuracy but they are computationally expensive and hard to deploy on any kind of vessel. Knowledge distillation (KD) is a model compression technique to obtain an accurate small model by transferring knowledge from big models [37]. KD has been widely used in face recognition tasks. For instance, ShrinkTeaNet [14] proposes an angular distillation loss that seeks to minimize the angle between teacher and student embedding vectors. KD has also been introduced to solve translation tasks and two sequence-level KD have been proven to be useful for translation [25]. The performance of the student model may be affected by the gap between the student and the teacher model. To effectively transfer knowledge to a smaller model, an intermediate teacher-assistant model is proposed to bridge the gap [37]. Another way to improve the knowledge transferring process is to use a grouped KD loss which consists of three parts proposed to filter out knowledge that is unrelated to facial identities [57]. To the best of our knowledge, KD has not been used in marine mammal recognition.
2.3 Comparison Models

This section will provide an in-depth overview of the models selected for the comparative study. By examining the architectures, strengths, and weaknesses of these models, we can lay the foundation for designing an optimized framework for the rapid and accurate classification of marine mammals’ sounds.

2.3.1 AlexNet

Starting with the commonly used CNNs, AlexNet is more or less the pioneer of CNN backbones. AlexNet is made up of five convolutional layers followed by three fully connected layers. The conventional activation function for a neuron’s output $f$ is $f(x) = \tanh(x)$ which is also the same as $f(x) = (1 + e^{-x})^{-1}$. The $\tanh$ function is a saturating non-linearity which is much slower than non-saturating non-linearity. The fully connected layers in AlexNet have an activation function of Rectified Linear Units (ReLU) that mitigates the vanishing gradient problem and speeds up training. Moreover, Local Response Normalization (LRN) is applied in the early layers to enhance the contrast between different features, improving the model’s generalizability. The response-normalized activity $b^i_{x,y}$ is defined by the following expression

$$b^i_{x,y} = \frac{a^i_{x,y}}{k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} (a^j_{x,y})^2}.$$  

The sum runs over $n$ “adjacent” kernel maps at the same spatial position, and $N$ is the total number of kernels in the layer. This sort of response normalization implements a form of lateral inhibition inspired by the type found in real neurons, creating competition for big activities amongst neuron outputs computed using different kernels. To be more precise, a pooling layer can be thought of as consisting of a grid of pooling units spaced $s$ pixels apart, each summarizing a neighborhood of size $z \times z$ centered at the location of the pooling unit. The conventional local pooling has $s = z$. 

13
whereas, AlexNet adapts $s < z$ to obtain an overlapping pooling.

The overall structure of AlexNet is shown in Figure 2.3. The first convolutional layer filters the $224 \times 224 \times 3$ input image with 96 kernels of size $11 \times 11 \times 3$ with a stride of 4 pixels. The output of the first convolutional layer is response-normalized, pooled, and passed to the second convolutional layer with filters of 256 kernels and size $5 \times 5 \times 48$. The next three convolutional layers are connected to one another without pooling or normalization. The third convolutional layer has 384 kernels of size $3 \times 3 \times 256$. The fourth convolutional layer has 384 kernels of size $3 \times 3 \times 192$, and the fifth convolutional layer has 256 kernels of size $3 \times 3 \times 192$. The fully-connected layers contain 4096 neurons each. AlexNet achieved the best performance for The ImageNet Large Scale Visual Recognition Challenge (ILSVRC). In conclusion, AlexNet is a breakthrough architecture in the computer vision field and it drives the development of advanced deep learning architectures and some of them will be described later.

![Figure 2.3: AlexNet architecture.](image)

### 2.3.2 VGG-16

VGG-16 is characterized by its architecture which is made up of 13 convolutional layers and 3 fully connected layers. VGG-16 uses 3x3 convolutional
2.3. **COMPARISON MODELS**

layers throughout the network and it aims to capture the complex features while restricting the model size. The hierarchy convolutional layers learn simple features at shallow layers and complex features at deep regions. A Max-Pooling is applied to every few convolutional layers to reduce the spatial dimensions so the computational complexity. The fully connected layers at the end of the network compute the class probabilities for the task. Similar to AlexNet, all the hidden layers contain the ReLU activation function; however, VGG does not contain any LRN. The existence of LRN in VGG will only increase memory consumption and computation time.

The straightforward design makes VGG-16 easy to implement and understand. However, its depth and the use of the small filters make VGG-16 computationally demanding and it can be slow in training and inference speed. Also, the VGG-16 architecture is large in size compared to other models so it may not be feasible to deploy in any kind of environment especially when the computational resources are limited. To sum up, VGG nets have been influential and competitive for many computer vision tasks but it has limitations that include computational complexity, employment difficulty, and overfitting.

![Figure 2.4: VGG architecture.](image)
2.3. COMPARISON MODELS

2.3.3 ResNet

ResNet is the abbreviation for ‘Residual Network’ and it is known for its strength in dealing an extreme-depth networks. A significant deep network usually leads to the gradient vanishing problem and ResNet tackles this problem by introducing residual blocks which is also known as the skip connection. Instead of learning the target output directly, the residual blocks learn the differences between the target output and input. The differences are added to the input to obtain the output, so the network learns the identity mapping. To better explain the residual blocks, we let $H(x)$ be an underlying mapping to be fit by the layers and $x$ be the inputs. Instead of approximating the $H(x)$ through the stacked layers, layers approximate a residual function $F(x) := H(x) - x$. Therefore, the original function becomes $F(x) + x$, improving the efficiency of the network. The network may find it difficult to approximate identity mappings through multiple non-linear layers and the identity mapping may not be optimal but this reformulation still helps the network to obtain better results. Every few stacked layers carry residual learning and the building block is defined as:

$$y = F(x, \{W_i\}) + x,$$  \hspace{1cm} (2.3)

where $F(x, \{W_i\})$ represents the residual mapping and $x$ is the input, and $y$ is the output. For example, in Figure 2.5, $\sigma$ denotes the ReLU function and $F + x$ is achieved through a shortcut connection and element-wise addition. The shortcut connections in ResNet add no burden to the computation. ResNet is designed to mitigate the gradient vanishing problem in such a way that the information is propagated from layer to layer without transformations. Furthermore, ResNet is inspired by VGG nets, and its convolutional layers are mostly made up of 3x3 filters and two design rules: the layers have the same number of filters as the output feature map size and the number of filters will be doubled if the feature map is halved to make the time complexity consistent. Hence, ResNet has few filters and lower complexity compared to VGG nets.

In addition, ResNet has a few variants that include ResNet-18 and
2.3. COMPARISON MODELS

ResNet-50 which are used for the comparison experiments. ResNet-50 has been proposed for marine mammal classification and then we decide to adopt ResNet18 to study their differences in efficiency and accuracy.

2.3.4 Data-efficient Image Transformers (DeiT)

DeiT is a vision transformer that is pre-trained using Imagenet and achieves state-of-the-art performance on different downstream tasks such as fine-grained classification and CIFAR-10. DeiT is a convolution-free transformer so the attention layers make up the major part of the network. The attention mechanism is associated with the concept of key and value pairs. A query, q vector is matched with a set of key, k vectors using inner products. The products will then be normalized and scaled to obtain the weights of k. The output of the attention maps is formulated as following:

\[
A(K, Q, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V,
\]  

(2.4)

where the \text{Softmax} function is applied to each row of the input and \(\sqrt{d}\) normalizes the attention maps. DeiT uses Multi-head self-attention layers (MSA) in the transformer which is defined by considering h attention ‘heads’. Each of the heads provides a sequence of size \(N \times d\) and the sequences are then arranged into \(N \times dh\) sequence which is further projected
2.3. COMPARISON MODELS

Figure 2.6: The ResNet-50 architecture. Adapted from [18].
by a linear layer into $N \times D$. There are a few advantages of MSA over the self-attention mechanism. The multiple heads allow the model to capture the information of the input from different aspects simultaneously. Each head learns different patterns and relationships of the features which helps the model learn the local and global dependencies effectively. MSA is also more robust to variability as different heads can interpret the various representations of identical information. DeiT’s transformer blocks are built upon the ViT model with a Feed-Forward Network (FNN) added on top of the MSA layer. The FFN consists of two linear layers with a Gaussian Error Linear Unit (GeLU) activation function in between. The first layer expands the feature dimension from $D$ to $4D$ and the second layer reverses it. Both MSA and GeLU work as residual operators with the aid of skip connections and layer normalization. The class token in Figure 2.7, is a trainable vector that is concatenated to the patch tokens at the beginning. The concatenated vectors are then passed into the transformer layers and projected to a linear layer for prediction. This class token is a concept that originated from Natural Language Processing (NLP) and the transformer will thus process batches of $(N + 1)$ tokens of dimension $D$ and only the class vector will be used for predicting the output. Such architecture restricts the attention layers to the spread of information between the patch tokens and the class tokens.

DeiT comes with a KD strategy that is specific to the transformer. This strategy involves twisting the teacher’s part, instead of measuring the Kullback-Leibler divergence between the teacher’s logits and the student’s logits, the KD loss replaces the teacher’s logits with the teacher’s hard labels:

$$L_{\text{DeiT}} = 0.5L_{\text{CE}}(\sigma(Z_s, y)) + 0.5L_{\text{CE}}(\sigma(Z_t), y_t),$$  \hspace{1cm} (2.5)

where $y_t = \arg\max_c Z_t(c)$ is the hard label predicted by the teacher model. This change is claimed to be better than the conventional one as it is parameter-free and conceptually simpler.

DeiT also introduces a new token, the distillation token. The distillation token works in a similar way to the class token as it interacts with patches
and class tokens through self-attention. The distillation token helps the DeiT student model learn from the teacher model and is complementary to the class embedding. The class tokens and distillation tokens will converge to similar vectors but not identical.

DeiT III is a proposal for a training recipe for the transformers that was introduced for ResNet-50 initially. The recipes include stochastic depth, Layerscale, 3-augmentation, and simple resized crop (SRC). Stochastic depth is a regularization technique for deep neural networks especially for CNN and ResNet. In a training with stochastic depth, a random subset of layers will be skipped for each input. It is similar to the Dropout mechanism which randomly drops neurons. The random skipping of layers encourages the network to learn to perform stably with different parts of the architecture disabled, making the network more robust. Layerscale is introduced to boost the convergence of the transformer network. For each layer, a scaling factor is introduced and the layer learns it to encourage a balanced flow of information and gradients. The use of 3-augmentation is inspired by self-supervised learning and it includes transformations: Grayscale, solarization, and Gaussian Blur. In addition to the 3 augmentations, DeiT III also includes color-jitter and horizontal flip. SRC is similar to the cropping technique proposed in AlexNet as it resizes the images so the smallest side matches the training resolution. Then apply a reflect padding on all sides and a square crop along the x-axis of of the image. SRC can reduce the discrepancy in apparent size and aspect ratio and it has a higher chance that the actual label is the same as the crop as compared to random cropping. It is also worth mentioning that DeiT variants DeiT-S and DeiT-Ti are smaller in size as compared to ResNet-50 and ResNet-18.

To sum up, DeiT III has proven to be the most powerful transformer up-to-date for the vision-related task that includes image classification and it is only pre-trained on ImageNet-1k which is relatively small compared to other transformers that are pre-trained on multiple datasets. Being the best model in the computer vision field, we include it to verify its performance in identifying marine mammals’ sound spectrograms with small data sizes.
2.3. COMPARISON MODELS

Figure 2.7: Deit structure.
2.3.5 Longformer

Longformer is a type of transformer that is created to solve the problem when the length of the sequences gets too long and the self-attention mechanism will require a significant amount of computation power which may not be feasible. The self-attention layers in Longformer scale linearly with the sequence length, so Longformer can handle sequences with various lengths. To further elaborate on how Longformer works, the Longformer sparsifies the full self-attention maps following an ‘attention pattern’ specifying pairs of input locations attending to one another. As shown in Figure 2.8. Longformer uses dilated sliding windows to determine the attention patterns which involves the use of a fixed-size window attention that surrounds each token. Stacking multiple layers of windowed attention to obtain a large receptive field, the top layers access all input locations and are able to build representations that incorporate information about the entire input. To further increase the receptive field without jeopardizing the model’s efficiency, the windows are dilated. Dilation creates gaps between the windows.

Figure 2.8: Architecture of a Longformer.
2.3. COMPARISON MODELS

Figure 2.9: Demonstration of different attention maps.

Windowed and dilated attention are not good enough to handle task-specific representations. Therefore, global attention is added to a few pre-selected input locations. The operation of attention is symmetric as a token is globally attended to all tokens across the sequence, and all tokens in the sequence attend to it. Figure 2.9 shows the difference of the attention maps. It is convenient to add inductive bias to the attention which is simpler than existing task-specific approaches that use complex networks to merge information. Recall that the equation of self-attention is 2.4, Longformer uses two sets of projections to compute attention scores of sliding window attention and global attention. The additional projections allow the model to be flexible to different types of attention.
2.3. COMPARISON MODELS

2.3.6 FNet

FNet is a small fast transformer encoder architecture that outperforms other transformers in speed. FNet replaces the attention layers in conventional transformers with Discrete Fourier Transform (DFT). Given a sequential input \(\{x_n\}\), the DFT layer generates a new representation \(X_k\) as a sum of input tokens in the following way:

\[
X_k = \sum_{n=0}^{N-2} x_ne^{-2\pi ink/N}, \quad 0 \leq k \leq N - 1.
\]  

This Fourier sublayer applies a 2D DFT to the embedding input as one along the sequence dimension, \(F_{\text{seq}}\) and the other one alone the hidden dimension, \(F_h\):

\[
y = R(F_{\text{seq}}(F_h(x))).
\]

The simplified representations of the tokens make FNet an efficient transformer.

![Figure 2.10: Structure of FNet.](image)
Chapter 3

Methodology

3.1 Overall Structure of The Framework

The overall flow of the structure is shown in Fig 3.2. The framework consists of two networks, the teacher network, and the student network. The teacher network is a deep and complex neural network. Whereas, the student network is much smaller. The teacher model will first be trained to obtain state-of-the-art accuracy through the model soup technique. Then the pre-trained teacher model will transfer the knowledge to the student model to help it learn the features.

Figure 3.1: The overall flow of XCFSMN.
3.2 Feature Extraction and Pre-processing

3.2.1 FBank Extraction

In this thesis, Log Mel-Filter Bank (Fbanks) is the feature used for model training. Compared to Mel-frequency cepstral coefficients (MFCCS), which is a feature commonly and conventionally used for automatic speech recognition, Fbanks are particularly well-suited for deep learning models as they preserve more spectral information. The feature extraction process begins with pre-emphasis applied to the audio signal. Pre-emphasis involves the use of a high-pass filter to balance the spectrum and reduce noise. Next, the continuous acoustic signal is segmented into discrete frames using a Hanning Window. The chosen sample rate for this process is 8kHz, with a window size of 512 samples and a window hop of 256 samples for each successive frame. In the third step of feature extraction, a Fast Fourier Transform (FFT) is applied to transform the segmented signal from the time domain to the frequency domain. The outcome of this transformation is an amplitude spectrum, often referred to as the Mel-Filter Bank, which captures essential frequency components of the audio signal. Finally, the last step involves the application of a logarithmic function to the Mel-Filter Bank, resulting in the Log Mel-Filter Bank or Fbanks. This transformation enhances the discriminative power of the feature while facilitating its use in deep-learning models for tasks such as marine mammal classification.

3.2.2 Cepstral Mean and Variance Normalization (CMVN)

The application of CMVN to Fbank features offers several advantages. This normalization process serves to standardize the features, resulting in a Gaussian distribution with values ranging from $-1$ to $1$. This standardization has several benefits for the training and performance of the model.

Firstly, it minimizes the effect of the amplitude of different frequency components. This is essential because large variations in feature magnitudes can lead to numerical instability during training, resulting in some features dominating the learning process while others become negligible.
3.2. FEATURE EXTRACTION AND PRE-PROCESSING

CMVN mitigates this issue by scaling all features to a comparable range. Secondly, normalization accelerates the model’s convergence speed. When input features are numerically close to each other, the optimization process tends to converge more rapidly. This can reduce the time and computational resources. Lastly, CMVN enhances the model’s robustness to noise in the input feature. Normalizing the features reduces variations in feature values caused by noise, making the model less sensitive to minor perturbations. The CMVN process is achieved through the following equation,

$$X_{normed} = \frac{X - \bar{X}}{\text{Var}(X)},$$

(3.1)

where $X_{normed}$ is calculated by each Fbank coefficient minus its mean and divided by its variance throughout the frames.
3.3 Teacher Model

The teacher model consists of three integral components: a pre-trained DenseNet network, a Time Delay Neural Network (TDNN), and a modified student model unit as shown in Figure 3.3. Each of these components plays a distinct role in the model’s overall functionality.

− Pre-trained DenseNet-121: DenseNet-121 is responsible for capturing and encoding the spectral information from the Fbank features, allowing the teacher model to learn intricate patterns and representations from the input data efficiently.

− TDNN: The TDNN is tasked with extracting the unique acoustic characteristics specific to different animals. It is designed to analyze the temporal aspects of the audio data, capturing the nuances and patterns that distinguish one species from another. The extracted patterns are known as the X-vectors.

− Modified Student Model Unit: This unit focuses on learning inter-frame context information. It complements the DenseNet-121 and TDNN by enhancing the model’s ability to understand the relationships between consecutive frames of audio data.

Collectively, this architectural design achieves state-of-the-art accuracy in marine mammal classification tasks. The teacher model transfers the knowledge to the student model, contributing to improved performance and facilitating efficient learning.

3.3.1 DenseNet-121

Incorporating a pre-trained CNN into the teacher model is a move aimed at enhancing feature extraction from the Fbank representations. CNNs excel at capturing spectro-temporal patterns, making them a valuable addition to the model’s architecture. For instance, the drastic visual differences between the FBanks of Minke Whales and Bowhead Whales can be effectively differentiated by CNN as shown in Figure 3.4. The selection of DenseNet-121 as
3.3. TEACHER MODEL

Figure 3.3: The structure of the teacher model.

Figure 3.4: FBank spectrogram of bowhead whale and minke whale.

the CNN backbone for our model is the result of a thorough experimentation process, where various CNN architectures were tested. There are a few
characteristics and advantages that will be elaborated. In the Dense Blocks, each layer obtains additional information from previous layers and passes on its own feature maps to the next layers. The $l^{th}$ layer has $l$ inputs that store feature maps from all previous convolutional layers. Subsequently, the output of $l^{th}$ will be passed to all layers after it:

$$x_l = H_l[x_0, X_1, \ldots, x_{l-1}]$$

(3.2)

The set $[x_0, X_1, \ldots, x_{l-1}]$ denotes the concatenated feature maps from the respective layer. There will be $\frac{L(L+1)}{2}$ connections in a network of $L$ layers. The dense connectivity pattern requires fewer parameters than the traditional CNNs.

ResNet presents such information through additive identity transformation but recent variations of ResNet suggest that many layers are redundant and can be dropped; whereas, DenseNet has narrow layers that only add a small set of feature maps to the network’s knowledge. $H_l$ refers to a composite function of three operations: batch normalization, ReLU, and a convolution layer with kernel size 3x3. However, if the size of the feature maps changes, the operation $H_l$ will fail, and down-sampling the features is an essential process in CNNs. To make down-sampling applicable, the network is divided into multiple dense blocks that are connected as shown in Figure 3.5. The layers in between the Dense Blocks are the transition layers which are made up of a batch normalization layer, a 1x1 convolutional layer, and a 2x2 average pooling layer. Each layer takes many more inputs than its output, hence, the introduction of a 1x1 convolution layer before each 3x3 convolution layer can reduce the size of feature maps, increasing computational efficiency. To further restrict the model’s size, a compression factor can be implemented at the transition layers. Assuming a dense block contains $m$ feature maps, the transition layer generates $\theta m$ output where $0 < \theta \leq 1$ is the compression factor. Hence, when $\theta = 1$, the feature maps remain the same.

To sum up, the densely connected neurons in Dense blocks receive full feature maps from the previous layer, reducing the gradient vanishing prob-
3.3. TEACHER MODEL

3.3. TEACHER MODEL

Such feature reuse nature also optimizes the gradient descent process. Further, the transition blocks between the dense blocks reduce the dimensions through convolution and pooling. This design of Dense blocks and transition blocks effectively compresses the model size and fosters efficient feature usage.

![DenseNet structure](image)

Figure 3.5: The structure of a DenseNet.

3.3.2 TDNN

The TDNN serves the role of an embedded X-vector extractor. X-vector [44] is a type of feature embedding that is commonly used in speaker recognition tasks such as speaker verification, speaker identification, and speaker diarization. The X-vector is a fixed-length feature representation of variable-length speech segments, such as utterances, that can capture short-term and long-term variations in speaker characteristics. By doing so, X-vectors reduce intra-speaker variability and expand inter-speaker variability. Implementing X-vectors can help the model differentiate cetaceans that produce sounds of similar frequencies.

The X-vector embedding model typically consists of five TDNN layers. The TDNN layers extract relevant features from the speech signal. These feature vectors are then subjected to a statistics pooling operation to create a consolidated and fixed-length X-vector representation. The X-vector is then used as input to a classifier, such as a support vector machine or neural network, to perform the speaker recognition task.

Conventionally, the X-vector extractor is trained isolated from the main
3.4. STUDENT MODEL (CC-FSMN)

classifier, we embed the X-vector model in the main model. This integration not only enhances efficiency but also reduces the overall model size. The embedded X-vector is concatenated to the output tensors from the CC-FSMN. This design choice enhances the model’s capacity to differentiate among cetaceans producing sounds with similar frequencies.

The Modified Student Model Unit has an identical architecture to the student model with a different number of neurons. The cFSMN [55] layer processes the contextual information that links up every frame in data and it is commonly used for automatic speech recognition. The XCA layers further strengthen the relationship between the frames. The justification for this step is that whales sing songs to communicate and the songs are like their languages. The details will be explained in the next section.

Figure 3.6: The structure of the teacher model.
3.4 Student Model (CC-FSMN)

3.4.1 Compact Feed-Forward Sequential Memory Network

Conventionally, long-term dependent data work ideally with Recurrent Neural Networks (RNNs) such as a Long Short Term Memory Network (LSTM). The nature of the network’s backpropagation through time [40] increases the RNNs’ computational complexity. Compared to Recurrent Neural Networks (RNNs), FNN has a simpler structure and learns faster and easier. Therefore, FNN with the ability to carry context information is preferred.

FSMN [56] is a new type of Feed-Forward Network (FNN) proposed in 2015. An illustration of an FSMN layer and the improved cFSMN layer is shown in Fig 3.8. FSMN is an FNN with memory blocks that encode the context information, allowing the transmission of long-term dependency in sequential data without requiring recurrent feedback. The calculation of memory block is defined as follows:

\[
\tilde{h}_i^\ell = \sum_{i=0}^{N_1} a_i^\ell \odot h_{i-1}^\ell + \sum_{j=0}^{N_2} c_j^\ell \odot h_{i+j}^\ell, \tag{3.3}
\]

where the \(a\) term encodes the context information up to previous \(N_1\) frames, and the \(c\) term encodes context information for future frames and \(\odot\) denotes matrix multiplication. The output \(\tilde{h}_i^\ell\) in equation (3.4) can be treated as
3.4. STUDENT MODEL (CC-FSMN)

Figure 3.8: The structures of FSMN (left) and cFSMN (right) networks.

A representation of the context information of the frames at time $t$. As a result, the context information is fed into the next layer using the output from the previous layer, as shown in the equation below:

$$h_{t+1}^l = \mathcal{F}(W^t h_t^l + \tilde{W}^t \tilde{h}_t^l + b^l),$$  \hspace{1cm} (3.4)

where $W$ and $b$ represent the standard weight matrix and bias vector, and $\tilde{W}$ denotes the weight matrix between the memory block and the next layer.

The cFSMN is a variant of FSMN architecture; it encodes the context information similarly to a standard FSMN. However, instead of using the full output $h_t^l$ from the previous layer, cFSMN projects the output to a linear layer. The memory block of cFSMN can be formulated as

$$\tilde{P}_t = P_t \sum_{i=0}^{N1} \tilde{a}_t^i \odot \tilde{P}_{t-i} + \sum_{j=0}^{N2} \tilde{c}_t^j \odot \tilde{P}_{t+j}. \hspace{1cm} (3.5)$$

The projected output $\tilde{P}_t$ in the memory block is then used for context
3.4. STUDENT MODEL (CC-FSMN)

information encoding. The output of the memory block is demonstrated by:

$$h_t^{l+1} = F(U^l P_t^l + b_t^{l+1}).$$  \hspace{1cm} (3.6)

To improve the cFSMN layer, we add an InstanceNorm layer after each cFSMN layer. The normalization process resolves the covariant shift that may occur in the network, which helps to speed up the training and stabilize the gradient descent process.

3.4.2 Cross-Covariance Attention (XCA)

In this framework, XCA is applied to the network to increase the robustness of the network and avoid the use of expensive computation of quadratic attention map. XCA is an attention mechanism proposed for an imager transformer that allows an image classifier to handle images of different resolutions. We believe XCA can be beneficial in the robustness of our classifier against noises.

Instead of a full pairwise interaction between the tokens by self-attention as shown in equation (2.4) and the following:

$$QK^T = XW_qW_kX^T,$$  \hspace{1cm} (3.7)

XCA derives the attention map using a Cross-Covariance matrix from the

Figure 3.9: The illustration of estimating attention in XCA.
projections of tokens using

\[ Axc(K, Q) = \text{Softmax}\left( \frac{\tilde{K}^T \tilde{Q}}{\tau} \right), \quad (3.8) \]

\[ K^T Q = W_k^T X^T X W_q. \quad (3.9) \]

The attention map is calculated by dividing the features into \( h \) groups which are similar to the idea of heads in multi-head attention. Then the calculation of the attention is restricted within the heads. XCA has two advantages: compared to self-attention, the complexity of the attention calculation is reduced by a factor of \( h \), the number of heads. Moreover, this block-diagonal version is easier to optimize, leading to improved results generally. The aforementioned advantages make XCA a better attention mechanism than self-attention for our objectives.

Conventionally, the XCA layer contains a dropout [45] operation that randomly drops out neurons in a layer to prevent the network from putting much weight on certain features. To avoid overfitting and improve the system’s robustness, we replace the dropout layer in the XCA with a DropKey [28] layer. Instead of blocking out neurons, DropKey drops the input Key units in the attention layer. The dropping of Keys before calculating the attention matrix can penalize weight peaks so regularize the weights. Hence, the DropKey mechanism has improved attention weight regularization, secures the patterns, and can avoid the local-bias problem. Practically, DropKey is done by randomly generating a mask matrix \( D \), and each element in the matrix \( D \) has a chance of \( d \) turning negative infinity. The output of the attention with a DropKey mechanism is formulated as:

\[ Axc(K, Q) = \text{Softmax}\left( d_j + \frac{\tilde{K}^T \tilde{Q}}{\tau} \right), \quad (3.10) \]

and \( d_j \sim \text{Bernoulli}(1 - dr) \).

To further decrease the size of the student model, we prune about 30% of the neurons from the cFSMN layer and the MLP layer through the global
3.5. LOSS FUNCTION

unstructured pruning (GUP) method. Instead of pruning an entire layer, GUP prunes neurons based on their importance scores, providing better granularity. GUP does not improve inference speed but it can compress a model well, making the model more adaptable to various capacity-restrained devices.

3.5 Loss Function

The framework involves two loss functions, a Label Smooth Cross-Entropy Loss, and a Cooling Knowledge Distillation Loss. The Cross-Entropy Loss function is commonly used for classification tasks. The model predicts the probability for each label $k$. $p(k|x) = \frac{e^{zk}}{\sum_{i=1}^{K} e^{zi}}$, where $z_i$ is the unnormalized log-probabilities. The Cross-Entropy loss function is then defined as $CE(i) = -\sum_{k=1}^{K} \log(p(k))q(k)$, and the $q(k)$ is the ground truth label distribution. The experiments adopt Label smoothing [47], which is a technique to regularize the model making it more robust by reducing the model’s confidence. Label smoothing is achieved through the introduction of a smoothing term $\epsilon$ to the Cross-Entropy function turning it into a function $(1 - \epsilon) \ast CE(i) + \frac{\epsilon}{K}$. Hence, instead of predicting 1 for the correct class and 0 for others, the model predicts $1 - \epsilon$. The student model’s evaluation partially depends on the Label Smooth Cross-Entropy Loss.

The loss used for knowledge distillation is called Cooling Knowledge Distillation Loss. It is a strategy we applied to mitigate the knowledge gap between the teacher model and the student model. The conventional knowledge distillation loss is formulated as:

$$L_{KD} = -T^2 \sum_{c} \sigma_{c}(\frac{Z_t}{T})\sigma_{c}(\frac{Z_s}{T})$$

The $L_{KD}$ is the cross-entropy between the teacher and student predictive probabilities $\sigma(Z_t)$ and $\sigma(Z_s)$. The $T$ term denotes the temperature hyperparameter which controls the discrepancy between two distributions. Increasing the $T$ will increase the information provided for the student model.
by the teacher model. It is wrongly assumed that a large pre-trained teacher model can always pair up with a small student model effectively [37]. Our proposed student model is approximately 18 times smaller than the teacher model. Hence, we believe the conventional KD loss may not be the best choice for this task. Moreover, most KD studies involve two networks of similar architecture but different layers and neurons. However, the proposed teacher model consists of a CNN backbone and a TDNN based on the student model. Hence, we propose to run a varying temperature strategy that reduces the knowledge gap caused by model size and architecture. The $T$ is replaced with $T_c = T_{co}(1 - \alpha)^n$ where $T_{co}$ is the initial temperature, $\alpha$ is the decreasing factor and $n$ is the number of decreases. Starting at $T_{co} = 20$, the temperature $T_c$ of the KD loss will decrease exponentially as the training process progresses at a rate of $\alpha = 0.1$. The criterion for triggering the decrease is when the early stopping counter reaches patience of 3. The rationale is explained as follows: a high temperature at the beginning allows the teacher model to transfer a large amount of knowledge to the student model. As time passes, the temperature decreases so the student model should learn the features more independently because of the difference between its and the teacher’s architecture. Therefore, the KD process becomes more effective.
Chapter 4

Experimental Setup

4.1 Dataset

Our study draws upon two distinct datasets to assess the robustness and performance of the systems. The first dataset, WMMSD contains approximately 2,000 audio recordings spanning over seven decades, encompassing the vocalizations of more than 60 marine mammal species. WMMSD presents challenges due to variations in recording equipment and setups over time. These fluctuations in data collection pose potential issues but also offer valuable insights into the system’s adaptability.

The second dataset employed is MobySound. The creation of MobySound was motivated by the need for automatic recognition tasks. MobySound holds a key advantage over WMMSD in terms of data consistency. Recordings within each species-specific sound set were collected using a constant configuration, ensuring the feature’s uniformity. Whereas, the variations in the sound collection configuration in the WMMSD allow us to test our system’s robustness and push the boundary that one classifier can achieve.

The two datasets share five common classes: Finback Whales, Humpback Whales, Blue Whales, Minke Whales, and Bowhead Whales. The audio length of the 5 classes is shown in Table 4.1. The sizes of the datasets show a significant difference. More, the classes are highly unbalanced which can generally lead to a biased model and poor generalization.

4.2 Experiment Details

The neural networks were implemented using Pytorch and trained using an RTX A6000 GPU. The choice of an input shape for the features as
4.2. EXPERIMENT DETAILS

Table 4.1: Audio length of the five classes in hours.

<table>
<thead>
<tr>
<th>Class</th>
<th>WMMSD</th>
<th>MobySound</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finback Whale</td>
<td>3.90</td>
<td>21.0</td>
</tr>
<tr>
<td>Humpback Whale</td>
<td>1.98</td>
<td>2.12</td>
</tr>
<tr>
<td>Blue Whale</td>
<td>0.12</td>
<td>10.20</td>
</tr>
<tr>
<td>Minke Whale</td>
<td>0.05</td>
<td>3.70</td>
</tr>
<tr>
<td>Bowhead Whale</td>
<td>0.39</td>
<td>1.10</td>
</tr>
</tbody>
</table>

256x256 was driven by the need to accommodate the input requirements of all the models used in our experiments. Regarding the audio data, a sample rate of 2kHz is sufficient to cover the frequencies of the 5 classes mentioned above. However, it is important to note that the soundtracks from the WMMSD dataset had relatively short durations, which made it challenging to create a frame number of 256 using the 2kHz sample rate. Therefore, we chose a sample rate of 8kHz for FBanks extraction.

To verify the efficiency and performance of XCFSMN, we made comparison experiments involving models as follows: CC-FSMN, AlexNet used in [30], ResNet50 used in [1], VGG-16, DenseNet used in [58], DeiT III [50], Longformer [54], and FNet [27]. In addition to ResNet50, we also run experiments on ResNet18, so every network in its smallest version will be tested for a comprehensive efficiency comparison.

The experiments conducted in this study encompass four distinct sets of data:

- WMMSD 12-Class Classification Task: In this task, the WMMSD dataset was employed for a 12-class classification problem. The dataset was structured to include the top 12 classes, ordered by sample size in descending order. This arrangement was chosen as classes beyond the top 12 had fewer than 100 samples, which was insufficient for meaningful training. The primary objective of this experiment was to assess the models’ capacity to handle highly imbalanced multi-class data.

- 5-Class Classification Tasks: A 5-class classification task for WMMSD
and MobySound each respectively. These two experiments were performed to verify the models’ ability to train data from an independent single source.

- Combined Dataset Classification Task: The data of 5 common classes from WMMSD and MobySound were combined for training and testing. This experiment aimed to assess the models’ robustness when confronted with the challenges of merging datasets with varying recording timing, target classes, and configurations. The task tests the models’ ability to perform well in a more complex and diverse dataset setting.

In all cases, the datasets were randomly split into training and test sets with a 7 to 3 ratio. It is worth noting that for the 5-class classification tasks, the dataset splitting occurred at the file level before the feature extraction step, maximizing the difference between the training set and test set. To avoid overfitting, we applied a cross-validation of 5 splits. We split the training set into a training set and a validation set with five different random states. The results were then averaged to obtain the final results. All the models were fine-tuned for the best comparison outcome.

### 4.3 Evaluation Metrics

To evaluate the efficiency of the models, training time and inference time were compared. Training time is the time taken for the model to finish 300 epochs or it triggers an early stop. Inference timing refers to the time taken for the model to predict the classes of the fixed-size test set. A shorter training time or inference time indicates improved efficiency. Also, inference time is more important than training time for the marine mammal classification tasks.

The Macro F1 score is typically used to evaluate the performance of the models as it takes into account both precision and recall. Instead of using accuracy, the F1 score can provide a more accurate measurement of the performance of the models given the datasets are highly unbalanced. F1
4.3. EVALUATION METRICS

Score is calculated through

\[
F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.
\]  

(4.1)

Precision in (4.1) is defined as the percentage of the correct labels out of all the predicted positives. Whereas, a model with high recall indicates that the model can find all the positive cases with some negative cases identified as positive. Since the objective is to detect and protect all kinds of marine mammals, a high recall is important.
Chapter 5

Results Comparison

5.1 Experiment Results

Table 5.1: Comparative results of XCFSMN and the baselines on efficiency. ‘TT’ denotes the training time in minutes and ‘IT’ denotes the inference time in seconds.

<table>
<thead>
<tr>
<th>Model</th>
<th># Params (M)</th>
<th>12-class WMMSD TT &amp; IT</th>
<th>5-class WMMSD TT &amp; IT</th>
<th>5-class MobySound TT &amp; IT</th>
<th>5-class Merged TT &amp; IT</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlexNet [30]</td>
<td>57.0</td>
<td>5.0</td>
<td>0.4</td>
<td>2.0</td>
<td>0.2</td>
</tr>
<tr>
<td>ResNet18 [18]</td>
<td>11.0</td>
<td>5.9</td>
<td>0.9</td>
<td>15.0</td>
<td>0.2</td>
</tr>
<tr>
<td>ResNet50 [1, 18]</td>
<td>23.5</td>
<td>2.7</td>
<td>0.6</td>
<td>15.6</td>
<td>1.7</td>
</tr>
<tr>
<td>VGG-16 [58]</td>
<td>134.0</td>
<td>3.5</td>
<td>3.9</td>
<td>10.0</td>
<td>1.0</td>
</tr>
<tr>
<td>DenseNet [58]</td>
<td>7.0</td>
<td>2.7</td>
<td>3.5</td>
<td>3.0</td>
<td>18.0</td>
</tr>
<tr>
<td>DeiT III [50]</td>
<td>21.6</td>
<td>3.2</td>
<td>16.3</td>
<td>6.0</td>
<td>4.7</td>
</tr>
<tr>
<td>Longformer [54]</td>
<td>16.9</td>
<td>47.8</td>
<td>8.4</td>
<td>14.0</td>
<td>11.0</td>
</tr>
<tr>
<td>FNet [27]</td>
<td>3.1</td>
<td>4.5</td>
<td>0.7</td>
<td>1.3</td>
<td>1.2</td>
</tr>
<tr>
<td>CC-FSMN [29]</td>
<td>4.5</td>
<td>2.7</td>
<td>0.32</td>
<td>0.7</td>
<td>1.0</td>
</tr>
<tr>
<td>Teacher (ours)</td>
<td>81.3</td>
<td>12.0</td>
<td>10.8</td>
<td>3.0</td>
<td>1.2</td>
</tr>
<tr>
<td>Student (ours)</td>
<td>3.15</td>
<td>5.8</td>
<td>0.32</td>
<td>1.0</td>
<td>0.1</td>
</tr>
</tbody>
</table>

The efficiency results are shown in Table 5.1 and the performance results are shown in Table 5.2, 5.3, 5.4, 5.5. Out of the 11 tested models, FNet is the smallest in model size followed by the proposed student model which is only approximately 1.6% larger. CC-FSMN [29] and the FNet [27] show excellent training speed in all the experiments. The teacher model takes much longer to train which is expected. The student model is a little slower in training speed. CC-FSMN and the student model have identical inference speeds and are dominant over other models in all experiments. This indicates that our proposed framework and CC-FSMN achieve state-of-the-art efficiency.

The teacher model shows absolute dominant results. The teacher model has the best precision, recall, and F1 Score for all the experiments. Through the temperature cooling knowledge distillation process, the student model
5.1. EXPERIMENT RESULTS

Table 5.2: Comparative results of 12-class WMMSD.

<table>
<thead>
<tr>
<th>Model</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlexNet [30]</td>
<td>0.594</td>
<td>0.735</td>
<td>0.612</td>
</tr>
<tr>
<td>ResNet18 [18]</td>
<td>0.782</td>
<td>0.846</td>
<td>0.806</td>
</tr>
<tr>
<td>ResNet50 [1, 18]</td>
<td>0.787</td>
<td>0.850</td>
<td>0.809</td>
</tr>
<tr>
<td>VGG-16 [58]</td>
<td>0.762</td>
<td>0.842</td>
<td>0.789</td>
</tr>
<tr>
<td>DenseNet [58]</td>
<td>0.705</td>
<td>0.810</td>
<td>0.735</td>
</tr>
<tr>
<td>DeiT III [50]</td>
<td>0.569</td>
<td>0.660</td>
<td>0.586</td>
</tr>
<tr>
<td>Longformer [54]</td>
<td>0.237</td>
<td>0.274</td>
<td>0.215</td>
</tr>
<tr>
<td>FNet [27]</td>
<td>0.120</td>
<td>0.225</td>
<td>0.120</td>
</tr>
<tr>
<td>CC-FSMN (ours)</td>
<td>0.785</td>
<td>0.828</td>
<td>0.787</td>
</tr>
<tr>
<td>Teacher (ours)</td>
<td>0.855</td>
<td>0.858</td>
<td>0.853</td>
</tr>
<tr>
<td>Student (ours)</td>
<td>0.816</td>
<td>0.837</td>
<td>0.820</td>
</tr>
</tbody>
</table>

Table 5.3: Comparative results of 5-class WMMSD.

<table>
<thead>
<tr>
<th>Model</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlexNet [30]</td>
<td>0.636</td>
<td>0.607</td>
<td>0.607</td>
</tr>
<tr>
<td>ResNet18 [18]</td>
<td>0.791</td>
<td>0.616</td>
<td>0.646</td>
</tr>
<tr>
<td>ResNet50 [1, 18]</td>
<td>0.791</td>
<td>0.616</td>
<td>0.646</td>
</tr>
<tr>
<td>VGG-16 [58]</td>
<td>0.891</td>
<td>0.823</td>
<td>0.783</td>
</tr>
<tr>
<td>DenseNet [58]</td>
<td>0.842</td>
<td>0.819</td>
<td>0.728</td>
</tr>
<tr>
<td>DeiT III [50]</td>
<td>0.781</td>
<td>0.704</td>
<td>0.660</td>
</tr>
<tr>
<td>Longformer [54]</td>
<td>0.472</td>
<td>0.247</td>
<td>0.275</td>
</tr>
<tr>
<td>FNet [27]</td>
<td>0.318</td>
<td>0.430</td>
<td>0.228</td>
</tr>
<tr>
<td>CC-FSMN (ours)</td>
<td>0.791</td>
<td>0.616</td>
<td>0.646</td>
</tr>
<tr>
<td>Teacher (ours)</td>
<td>0.946</td>
<td>0.828</td>
<td>0.827</td>
</tr>
<tr>
<td>Student (ours)</td>
<td>0.900</td>
<td>0.824</td>
<td>0.791</td>
</tr>
</tbody>
</table>

has significant improvements from the CC-FSMN. There is a minimum improvement of 0.064 in the absolute F1 Score and a maximum of 0.145. Despite the student model not obtaining as high an F1 Score as the teacher model, it still outperforms all other models given it has almost the smallest size.

The CNNs have similar performance except for AlexNet which shows significant drops in 12-class WMMSD and the merged dataset. It is observed
5.1. EXPERIMENT RESULTS

Table 5.4: Comparative results of 5-class Moby.

<table>
<thead>
<tr>
<th>Model</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlexNet [30]</td>
<td>0.865</td>
<td>0.884</td>
<td>0.863</td>
</tr>
<tr>
<td>ResNet18 [18]</td>
<td>0.950</td>
<td>0.873</td>
<td>0.880</td>
</tr>
<tr>
<td>ResNet50 [1, 18]</td>
<td>0.928</td>
<td>0.845</td>
<td>0.846</td>
</tr>
<tr>
<td>VGG-16 [58]</td>
<td>0.860</td>
<td>0.833</td>
<td>0.827</td>
</tr>
<tr>
<td>DenseNet [58]</td>
<td>0.957</td>
<td>0.897</td>
<td>0.911</td>
</tr>
<tr>
<td>DeiT III [50]</td>
<td>0.892</td>
<td>0.860</td>
<td>0.896</td>
</tr>
<tr>
<td>Longformer [54]</td>
<td>0.657</td>
<td>0.705</td>
<td>0.645</td>
</tr>
<tr>
<td>FNet [27]</td>
<td>0.275</td>
<td>0.561</td>
<td>0.283</td>
</tr>
<tr>
<td>CC-FSMN (ours)</td>
<td>0.970</td>
<td>0.898</td>
<td>0.922</td>
</tr>
<tr>
<td>Teacher (ours)</td>
<td><strong>0.981</strong></td>
<td><strong>0.991</strong></td>
<td><strong>0.986</strong></td>
</tr>
<tr>
<td>Student (ours)</td>
<td>0.970</td>
<td>0.977</td>
<td>0.973</td>
</tr>
</tbody>
</table>

Table 5.5: Comparative results of 5-class merged.

<table>
<thead>
<tr>
<th>Model</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlexNet [30]</td>
<td>0.867</td>
<td>0.800</td>
<td>0.782</td>
</tr>
<tr>
<td>ResNet18 [18]</td>
<td>0.934</td>
<td>0.809</td>
<td>0.840</td>
</tr>
<tr>
<td>ResNet50 [1, 18]</td>
<td>0.904</td>
<td>0.828</td>
<td>0.842</td>
</tr>
<tr>
<td>VGG-16 [58]</td>
<td>0.784</td>
<td>0.801</td>
<td>0.790</td>
</tr>
<tr>
<td>DenseNet [58]</td>
<td>0.916</td>
<td>0.830</td>
<td>0.869</td>
</tr>
<tr>
<td>DeiT III [50]</td>
<td>0.928</td>
<td>0.740</td>
<td>0.787</td>
</tr>
<tr>
<td>Longformer [54]</td>
<td>0.668</td>
<td>0.594</td>
<td>0.627</td>
</tr>
<tr>
<td>FNet [27]</td>
<td>0.261</td>
<td>0.296</td>
<td>0.222</td>
</tr>
<tr>
<td>CC-FSMN (ours)</td>
<td>0.893</td>
<td>0.817</td>
<td>0.828</td>
</tr>
<tr>
<td>Teacher (ours)</td>
<td><strong>0.976</strong></td>
<td><strong>0.890</strong></td>
<td><strong>0.925</strong></td>
</tr>
<tr>
<td>Student (ours)</td>
<td>0.944</td>
<td>0.890</td>
<td>0.910</td>
</tr>
</tbody>
</table>

that all the transformer models perform poorly. The poor performance can be attributed to the small dataset size as the vision transformers require large datasets to demonstrate their strength for image classification tasks [46]. DeiT can keep up in the 5-class experiments but fails for the 12-class because the 12-class WMMSD has a really limited sample size for the minority class. Longformer performs poorly since it was designed for long-sequence languages. More, the replacement of the attention mechanism with
the Fourier Mixing Layer in FNet restricts the model from capturing long-sequence context information reducing the model’s ability to classify the signal.

To better illustrate the ranking of the models. We performed The Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS). The weights of precision, recall, and training time are assigned with a value of 0.1, and a value of 0.35 is assigned to the F1 Score and inference time since identifying the animals quickly and accurately is prioritized. The average performance scores indicate that the student model is the best choice followed by CC-FSMN and then ResNet18.

In addition, we compare the results of the aforementioned photo approaches [32, 41]. The mean F1 score achieved by the model trained using aerial images in [32] is 0.4059 and classification using Commerson’s dolphins’ photos in [41] is 0.883. XCFSMN shows a better F1 score and also it is capable of identifying multiple species. Therefore, marine mammal classification based on acoustic signals with XCFSMN is better than the same classification problem based on image processing with deep learning.

Looking at the confusion matrix of a 5-class experiment run with the student model in Figure 5.1, the model has difficulty distinguishing the spectrograms of the Finback Whale and the Blue Whale. Perhaps this can be explained by Figure 5.2, both the Fin Whale and Blue Whale spectrograms contain a large blank space caused by a sampling rate that was too high. The large blank space caused the meaningful part to be small, making it difficult for the model to find the differences between them.

### 5.2 Ablation Study

We conducted an ablation study for the student model shown in Table 5.6 by comparing efficiency and performance using the 5-class merged dataset. The first row shows the results for a CC-FSMN [29] and behaves as the baseline. The addition of Instance Normalization to the output of cFSMN increases the model’s precision and is able to maintain the recall. Replacing Dropout with DropKey further increases the precision but reduces the recall.
5.2. ABLATION STUDY

Figure 5.1: Confusion matrix for student model’s results on combined dataset.
5.2. ABLATION STUDY

Figure 5.2: Spectrograms of three whale species.

Table 5.6: Results of the ablation study on the student model. ‘TT’ denotes the training time in minutes and ‘IT’ denotes the inference time in seconds.

<table>
<thead>
<tr>
<th>Structure</th>
<th>InstanceNorm</th>
<th>DropKey</th>
<th>KD</th>
<th>TT</th>
<th>IT</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>×</td>
<td>×</td>
<td>×</td>
<td></td>
<td>2.5</td>
<td>0.32</td>
<td>0.893</td>
<td>0.817</td>
<td>0.828</td>
</tr>
<tr>
<td>✓</td>
<td>×</td>
<td>×</td>
<td></td>
<td>1.8</td>
<td>0.32</td>
<td>0.939</td>
<td>0.814</td>
<td>0.839</td>
</tr>
<tr>
<td>×</td>
<td>✓</td>
<td>×</td>
<td></td>
<td>1.5</td>
<td>0.38</td>
<td>0.954</td>
<td>0.805</td>
<td>0.844</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td></td>
<td>1.7</td>
<td>0.38</td>
<td>0.951</td>
<td>0.817</td>
<td>0.850</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>17</td>
<td>0.33</td>
<td>0.944</td>
<td>0.890</td>
<td>0.910</td>
</tr>
</tbody>
</table>

The combination of Instance Normalization and DropKey boosts the student model’s precision without affecting its recall. Finally, the introduction of temperature-cooling KD increases the recall and F1 Score, achieving state-of-the-art.

The second ablation study on the KD temperature is presented in Table 6.1. The experiments were conducted on different fixed temperatures ranging from 1 to 40 and the best one was selected for comparison. We also include a temperature-increasing strategy that works in a reverse way to our strategy. The results indicate that fixing temperature cannot fully exploit the benefits of KD. Either increasing or decreasing the temperature parameter during the KD process allows the student model to acquire knowledge from the teacher more effectively. Both warming and cooling strategies have similar training times, but the cooling strategy shows improved recall and F1 Score.
5.2. ABLATION STUDY

Table 5.7: Results of the ablation study on the KD temperature strategy. ‘TT’ denotes the training time in minutes and ‘IT’ denotes the inference time in seconds.

<table>
<thead>
<tr>
<th>Temperature Strategy</th>
<th>TT</th>
<th>IT</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fix T = 10</td>
<td>1.5</td>
<td>0.3</td>
<td>0.918</td>
<td>0.836</td>
<td>0.854</td>
</tr>
<tr>
<td>Warming</td>
<td>3.5</td>
<td>0.3</td>
<td>0.943</td>
<td>0.859</td>
<td>0.887</td>
</tr>
<tr>
<td>Cooling</td>
<td>3.5</td>
<td>0.3</td>
<td>0.944</td>
<td>0.890</td>
<td>0.910</td>
</tr>
</tbody>
</table>

To provide a more intuitive result, we generated the t-distributed stochastic neighbor embedding (t-SNE) to visualize the high-dimensional information from the neurons of the output layer. Fig 5.3 shows that FNet is unable to differentiate most of the classes. Transformer DeiT can isolate the major classes but handles the minorities poorly. The teacher model demonstrates the best capability to isolate different classes. The student model isolates class 0 and class 6 better than CC-FSMN. Therefore, we can claim that the proposed framework can capture minor differences between classes better and is more robust than CC-FSMN.
5.2. ABLATION STUDY

Figure 5.3: Visualization of features using a t-SNE plot for 12-class Watkins experiment.
Chapter 6

Conclusion and Future Work

This thesis proposed the XCFSMN framework that trains a small and fast student model using a pre-trained teacher model with a temperature-cooling strategy. The framework exhibits superior efficiency and performance compared to CNN backbones and transformers for marine mammal sound classification. Such an acoustic approach has shown better accuracy than the approach based on images. XCFSMN has significant improvements in performance and similar efficiency compared to CC-FSMN. Moreover, the model’s pruning also enhances its deployability across various devices.

To further improve the classifier, future work can focus on data preprocessing techniques rather than model architecture adjustments. Specifically, adopting a diffusion model [19] to denoise the data and generate balanced datasets. It is worth mentioning that XCFSMN’s applicability extends beyond marine mammal sound classification. This framework’s concept and methodology can be adapted for other acoustic signal classification tasks, such as speech sentiment recognition and speaker identification, offering versatility and broader potential applications in the field of acoustic signal analysis.
Bibliography


Bibliography


Bibliography


Bibliography


[37] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, Hassan Ghasemzadeh, and De Shaw. Improved knowledge distillation via teacher assistant. 2019. → pages 12, 38


[47] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, and Jonathon Shlens. Rethinking the inception architecture for computer vision. 2016. → pages 37


[53] Kang You, Kele Xu, Boqing Zhu, Ming Feng, Dawei Feng, Bo Liu, Tian Gao, and Bo Ding. Masked modeling-based audio representation for acm multimedia 2022 computational paralinguistics challenge. pages


[56] Shiliang Zhang, Cong Liu, Hui Jiang, Si Wei, Lirong Dai, and Yu Hu. Feedforward sequential memory networks: A new structure to learn long-term dependency. 12 2015. → pages 33


Table 6.1: TOPSIS score of 5-class Moby data.

<table>
<thead>
<tr>
<th>Model</th>
<th>Norm Precision</th>
<th>Norm Recall</th>
<th>Norm F1</th>
<th>Norm TT</th>
<th>Norm IT</th>
<th>S+</th>
<th>S-</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlexNet</td>
<td>0.02987</td>
<td>0.03111</td>
<td>0.10778</td>
<td>0.01022</td>
<td>0.00423</td>
<td>0.0187</td>
<td>0.25696</td>
<td>0.93496</td>
</tr>
<tr>
<td>ResNet-18</td>
<td>0.03280</td>
<td>0.03072</td>
<td>0.10990</td>
<td>0.00876</td>
<td>0.01269</td>
<td>0.0173</td>
<td>0.25027</td>
<td>0.93534</td>
</tr>
<tr>
<td>resnet-50</td>
<td>0.03204</td>
<td>0.02973</td>
<td>0.10565</td>
<td>0.01022</td>
<td>0.03383</td>
<td>0.0355</td>
<td>0.22922</td>
<td>0.86567</td>
</tr>
<tr>
<td>vgg-16</td>
<td>0.02969</td>
<td>0.02931</td>
<td>0.10328</td>
<td>0.04673</td>
<td>0.05709</td>
<td>0.0718</td>
<td>0.19873</td>
<td>0.73458</td>
</tr>
<tr>
<td>DenseNet</td>
<td>0.03304</td>
<td>0.03156</td>
<td>0.11377</td>
<td>0.01752</td>
<td>0.04229</td>
<td>0.0497</td>
<td>0.22210</td>
<td>0.84107</td>
</tr>
<tr>
<td>Deit3</td>
<td>0.03090</td>
<td>0.03026</td>
<td>0.11190</td>
<td>0.07594</td>
<td>0.24316</td>
<td>0.2501</td>
<td>0.07204</td>
<td>0.22359</td>
</tr>
<tr>
<td>Longformer</td>
<td>0.02268</td>
<td>0.02481</td>
<td>0.08055</td>
<td>0.01752</td>
<td>0.09727</td>
<td>0.1044</td>
<td>0.16198</td>
<td>0.60797</td>
</tr>
<tr>
<td>Fnet</td>
<td>0.01274</td>
<td>0.02055</td>
<td>0.04284</td>
<td>0.00292</td>
<td>0.00846</td>
<td>0.0843</td>
<td>0.24580</td>
<td>0.74447</td>
</tr>
<tr>
<td>CC-FSMN</td>
<td>0.03373</td>
<td>0.03160</td>
<td>0.11515</td>
<td>0.00438</td>
<td>0.00423</td>
<td>0.0087</td>
<td>0.26077</td>
<td>0.96750</td>
</tr>
<tr>
<td>Teacher</td>
<td>0.03387</td>
<td>0.03487</td>
<td>0.12314</td>
<td>0.03213</td>
<td>0.21779</td>
<td>0.2155</td>
<td>0.09830</td>
<td>0.31321</td>
</tr>
<tr>
<td>Student</td>
<td>0.03349</td>
<td>0.03438</td>
<td>0.12151</td>
<td>0.00949</td>
<td>0.00423</td>
<td>0.0068</td>
<td>0.26137</td>
<td>0.97465</td>
</tr>
</tbody>
</table>
## Appendix A: Tables

Table 6.2: TOPSIS score of 5-class Watkins data.

<table>
<thead>
<tr>
<th>Model</th>
<th>Norm Precision</th>
<th>Norm Recall</th>
<th>Norm F1</th>
<th>Norm TT</th>
<th>S+</th>
<th>S−</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlexNet</td>
<td>0.024624494</td>
<td>0.027156983</td>
<td>0.098392909</td>
<td>0.012133865</td>
<td>0.002890465</td>
<td>0.040280719</td>
<td>0.145914568</td>
</tr>
<tr>
<td>Resnet18</td>
<td>0.030625746</td>
<td>0.027924831</td>
<td>0.104714694</td>
<td>0.00666933</td>
<td>0.00578903</td>
<td>0.031860124</td>
<td>0.147129696</td>
</tr>
<tr>
<td>Resnet50</td>
<td>0.030625746</td>
<td>0.027924831</td>
<td>0.104714694</td>
<td>0.01313755</td>
<td>0.017342791</td>
<td>0.035698585</td>
<td>0.136196555</td>
</tr>
<tr>
<td>vgg16</td>
<td>0.034497522</td>
<td>0.037308662</td>
<td>0.12692199</td>
<td>0.016380718</td>
<td>0.028904651</td>
<td>0.030727925</td>
<td>0.138501422</td>
</tr>
<tr>
<td>DenseNet</td>
<td>0.032561634</td>
<td>0.037423132</td>
<td>0.118006652</td>
<td>0.02790789</td>
<td>0.023123721</td>
<td>0.036934778</td>
<td>0.136833566</td>
</tr>
<tr>
<td>Deit3</td>
<td>0.030238568</td>
<td>0.031940403</td>
<td>0.106984053</td>
<td>0.036401595</td>
<td>0.13581861</td>
<td>0.104284783</td>
<td>0.56431583</td>
</tr>
<tr>
<td>Longformer</td>
<td>0.018274784</td>
<td>0.011971312</td>
<td>0.044576869</td>
<td>0.084937056</td>
<td>0.317951464</td>
<td>0.339423652</td>
<td>0.188501422</td>
</tr>
<tr>
<td>Fnet</td>
<td>0.022909124</td>
<td>0.032997755</td>
<td>0.046525077</td>
<td>0.00182008</td>
<td>0.003356987</td>
<td>0.093024087</td>
<td>0.135866699</td>
</tr>
<tr>
<td>CC-FSMN</td>
<td>0.030625746</td>
<td>0.027924831</td>
<td>0.104714694</td>
<td>0.004550199</td>
<td>0.014273829</td>
<td>0.020540188</td>
<td>0.125312884</td>
</tr>
<tr>
<td>Teacher</td>
<td>0.036626998</td>
<td>0.037535995</td>
<td>0.128218766</td>
<td>0.006066933</td>
<td>0.002890465</td>
<td>0.00743597</td>
<td>0.161843592</td>
</tr>
<tr>
<td>Student</td>
<td>0.034845982</td>
<td>0.037353995</td>
<td>0.128218766</td>
<td>0.006066933</td>
<td>0.002890465</td>
<td>0.00743597</td>
<td>0.161843592</td>
</tr>
</tbody>
</table>

Table 6.3: TOPSIS score of 5-class merged data.

<table>
<thead>
<tr>
<th>Model</th>
<th>Norm Precision</th>
<th>Norm Recall</th>
<th>Norm F1</th>
<th>Norm TT</th>
<th>S+</th>
<th>S−</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlexNet</td>
<td>0.030823731</td>
<td>0.031257854</td>
<td>0.104679055</td>
<td>0.00470354</td>
<td>0.020549188</td>
<td>0.125312884</td>
<td>0.859172056</td>
</tr>
<tr>
<td>Resnet18</td>
<td>0.033205726</td>
<td>0.031609505</td>
<td>0.112442975</td>
<td>0.013065389</td>
<td>0.028547659</td>
<td>0.024710161</td>
<td>0.118970805</td>
</tr>
<tr>
<td>Resnet50</td>
<td>0.032139161</td>
<td>0.032351879</td>
<td>0.117180696</td>
<td>0.01646239</td>
<td>0.078586061</td>
<td>0.070559269</td>
<td>0.099729296</td>
</tr>
<tr>
<td>vgg16</td>
<td>0.027872901</td>
<td>0.031996296</td>
<td>0.105789941</td>
<td>0.04422323</td>
<td>0.138032921</td>
<td>0.130422158</td>
<td>0.091189008</td>
</tr>
<tr>
<td>DenseNet</td>
<td>0.032567877</td>
<td>0.034309024</td>
<td>0.116249395</td>
<td>0.01695826</td>
<td>0.092779891</td>
<td>0.084051202</td>
<td>0.101675765</td>
</tr>
<tr>
<td>Deit3</td>
<td>0.032992413</td>
<td>0.029915151</td>
<td>0.10543859</td>
<td>0.00101079</td>
<td>0.092779891</td>
<td>0.10341757</td>
<td>0.084179692</td>
</tr>
<tr>
<td>Longformer</td>
<td>0.037488494</td>
<td>0.032389957</td>
<td>0.083930649</td>
<td>0.049648479</td>
<td>0.235518184</td>
<td>0.234037302</td>
<td>0.15416721</td>
</tr>
<tr>
<td>Fnet</td>
<td>0.009270116</td>
<td>0.015654866</td>
<td>0.029710712</td>
<td>0.00783923</td>
<td>0.01742287</td>
<td>0.10045762</td>
<td>0.093572246</td>
</tr>
<tr>
<td>CC-FSMN</td>
<td>0.031748888</td>
<td>0.039220883</td>
<td>0.110836647</td>
<td>0.006532995</td>
<td>0.010705372</td>
<td>0.014781345</td>
<td>0.130787621</td>
</tr>
<tr>
<td>Teacher</td>
<td>0.034698918</td>
<td>0.034774363</td>
<td>0.123811213</td>
<td>0.01063589</td>
<td>0.157012123</td>
<td>0.146923139</td>
<td>0.127975353</td>
</tr>
<tr>
<td>Student</td>
<td>0.035561248</td>
<td>0.034774363</td>
<td>0.121843223</td>
<td>0.03156934</td>
<td>0.010705372</td>
<td>0.03065999</td>
<td>0.1304355</td>
</tr>
</tbody>
</table>

60
Appendix A: Tables

Table 6.4: TOPSIS score of 12-class Watkins data.

<table>
<thead>
<tr>
<th>Model</th>
<th>Norm Precision</th>
<th>Norm Recall</th>
<th>Norm F1</th>
<th>Norm TT</th>
<th>Norm IT</th>
<th>$S_+$</th>
<th>$S_-$</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlexNet</td>
<td>0.0263835</td>
<td>0.02993865</td>
<td>0.093777835</td>
<td>0.007126721</td>
<td>0.038446595</td>
<td>0.254422254</td>
<td>0.869090401</td>
<td></td>
</tr>
<tr>
<td>Resnet18</td>
<td>0.03473584</td>
<td>0.03146065</td>
<td>0.12292334</td>
<td>0.013589188</td>
<td>0.011911235</td>
<td>0.254583896</td>
<td>0.955350109</td>
<td></td>
</tr>
<tr>
<td>Resnet50</td>
<td>0.03490917</td>
<td>0.034622936</td>
<td>0.123359787</td>
<td>0.01931336</td>
<td>0.039499692</td>
<td>0.237803848</td>
<td>0.859321958</td>
<td></td>
</tr>
<tr>
<td>Vgg16</td>
<td>0.0338455</td>
<td>0.034287074</td>
<td>0.120310101</td>
<td>0.008047233</td>
<td>0.013589188</td>
<td>0.0921518066</td>
<td>0.187907458</td>
<td>0.670941199</td>
</tr>
<tr>
<td>Densenet</td>
<td>0.03113175</td>
<td>0.03299322</td>
<td>0.11207595</td>
<td>0.009358287</td>
<td>0.051834169</td>
<td>0.069562783</td>
<td>0.212703058</td>
<td>0.77890257</td>
</tr>
<tr>
<td>Deit3</td>
<td>0.025273083</td>
<td>0.026883622</td>
<td>0.089355701</td>
<td>0.046820267</td>
<td>0.238347577</td>
<td>0.241997861</td>
<td>0.675804333</td>
<td>0.238750969</td>
</tr>
<tr>
<td>Longformer</td>
<td>0.01052675</td>
<td>0.01160695</td>
<td>0.032784121</td>
<td>0.054867501</td>
<td>0.167906465</td>
<td>0.199984079</td>
<td>0.670444276</td>
<td>0.271219133</td>
</tr>
<tr>
<td>FNet</td>
<td>0.00553</td>
<td>0.00964895</td>
<td>0.018289814</td>
<td>0.0065841</td>
<td>0.01027873</td>
<td>0.119392105</td>
<td>0.232575763</td>
<td>0.66078692</td>
</tr>
<tr>
<td>CC-FSMN</td>
<td>0.034867084</td>
<td>0.037269173</td>
<td>0.120005132</td>
<td>0.001463133</td>
<td>0.004698848</td>
<td>0.00177573</td>
<td>0.262579213</td>
<td>0.97075355</td>
</tr>
<tr>
<td>Teacher</td>
<td>0.03797625</td>
<td>0.034948799</td>
<td>0.130065132</td>
<td>0.01757576</td>
<td>0.158586125</td>
<td>0.154279033</td>
<td>0.14976377</td>
<td>0.47881246</td>
</tr>
<tr>
<td>Student</td>
<td>0.036244</td>
<td>0.034093409</td>
<td>0.125037114</td>
<td>0.008486173</td>
<td>0.004698848</td>
<td>0.009715864</td>
<td>0.26362824</td>
<td>0.97877583</td>
</tr>
</tbody>
</table>

Table 6.5: Overall TOPSIS score.

<table>
<thead>
<tr>
<th>Model</th>
<th>Moby 5</th>
<th>Watkins 5</th>
<th>Merged 5</th>
<th>Watkins 12</th>
<th>Averaged Score</th>
<th>Ranking</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlexNet</td>
<td>0.934964074</td>
<td>0.783664134</td>
<td>0.859172056</td>
<td>0.86906401</td>
<td>0.861176666</td>
<td>4</td>
</tr>
<tr>
<td>Resnet18</td>
<td>0.935344641</td>
<td>0.821946983</td>
<td>0.827995353</td>
<td>0.955350109</td>
<td>0.885159271</td>
<td>3</td>
</tr>
<tr>
<td>Resnet50</td>
<td>0.865674213</td>
<td>0.792501401</td>
<td>0.58706386</td>
<td>0.859321958</td>
<td>0.775800989</td>
<td>5</td>
</tr>
<tr>
<td>Vgg16</td>
<td>0.734575863</td>
<td>0.818526587</td>
<td>0.411296057</td>
<td>0.670911999</td>
<td>0.658834926</td>
<td>7</td>
</tr>
<tr>
<td>Densenet</td>
<td>0.841074588</td>
<td>0.787448177</td>
<td>0.547436013</td>
<td>0.778396257</td>
<td>0.738588759</td>
<td>6</td>
</tr>
<tr>
<td>Deit3</td>
<td>0.223594315</td>
<td>0.316072208</td>
<td>0.440555672</td>
<td>0.238756969</td>
<td>0.304694814</td>
<td>11</td>
</tr>
<tr>
<td>Longformer</td>
<td>0.607974055</td>
<td>0.35709552</td>
<td>0.397128847</td>
<td>0.271219133</td>
<td>0.408354497</td>
<td>10</td>
</tr>
<tr>
<td>FNet</td>
<td>0.744471029</td>
<td>0.593800413</td>
<td>0.487545289</td>
<td>0.66078692</td>
<td>0.621650913</td>
<td>8</td>
</tr>
<tr>
<td>CC-FSMN</td>
<td>0.967497744</td>
<td>0.826113932</td>
<td>0.898458132</td>
<td>0.957075355</td>
<td>0.912287638</td>
<td>2</td>
</tr>
<tr>
<td>Teacher</td>
<td>0.331321065</td>
<td>0.812596383</td>
<td>0.465715436</td>
<td>0.487881246</td>
<td>0.519850906</td>
<td>9</td>
</tr>
<tr>
<td>Student</td>
<td>0.974653232</td>
<td>0.956072843</td>
<td>0.810393835</td>
<td>0.978778581</td>
<td>0.929974395</td>
<td>1</td>
</tr>
</tbody>
</table>