Deep Learning for Orca Call Type Identification – A Fully Unsupervised Approach

Christian Bergler[1], *Manuel Schmitt*[1], *Rachael Xi Cheng*[2], *Andreas Maier*[1], *Volker Barth*[3], *Elmar Nöth*[1]

[1]Friedrich-Alexander-University Erlangen-Nuremberg, Department of Computer Science – Pattern Recognition Lab, Martensstr. 3, 91058 Erlangen, Germany
[2]Leibniz Institute for Zoo and Wildlife Research (IZW) in the Forschungsverbund Berlin e.V., Alfred-Kowalke-Straße 17, 10315 Berlin, Germany
[3]Anthro-Media, Nansenstr. 19, 12047 Berlin, Germany

{christian.bergler, elmar.noeth}@fau.de

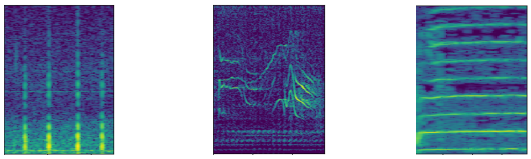
Abstract

Call type classification is an important instrument in bioacoustic research investigating group-specific vocal repertoire, behavioral patterns, and cultures of different animal groups. There is a growing need using robust machine-based techniques to replace human classification due to its advantages in handling large datasets, delivering consistent results, removing perceptual-based classification, and minimizing human errors. The current work is the first adopting a two-stage fully unsupervised approach on previous machine-segmented orca data to identify orca sound types using deep learning together with one of the largest bioacoustic datasets – the Orchive. The proposed methods include: (1) unsupervised feature learning using an undercomplete ResNet18-autoencoder trained on machine-annotated data, and (2) spectral clustering utilizing compressed orca feature representations. An existing human-labeled orca dataset was clustered, including 514 signals distributed over 12 classes. This two-stage fully unsupervised approach is an initial study to (1) examine machine-generated clusters against human-identified orca call type classes, (2) compare supervised call type classification versus unsupervised call type clustering, and (3) verify the general feasibility of a completely unsupervised approach based on machine-labeled orca data resulting in a major progress within the research field of animal linguistics, by deriving a much deeper understanding and facilitating totally new insights and opportunities.

Index Terms: orca, call type, unsupervised, deep learning, clustering

1. Introduction

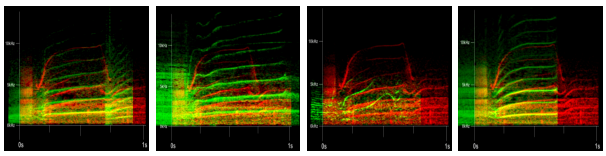
Call type classification is an important instrument to identify species and to track movements of animal groups and therefore to identify habitat usage. It is also important to further study group-specific vocal repertoire, behaviors, and cultures of different animal groups. With an increasing amount of passive acoustic monitoring data comes a growing need for using robust automatic techniques to replace human classification. Apart from processing large amounts of data in short time, those techniques offer advantages of delivering consistent results and are replicable in subsequent studies. The largest member of the dolphin family - the Orca (*Orcinus Orca*) - is one of several species with relatively well-studied and complex vocal cultures [1]. The acoustic behavior of killer whales has been extensively studied on the resident fish-eating orcas in the Northeast Pacific. Besides echolocation clicks and whistles, orcas produce a number of different, group-specific, and social sounds with distinct frequency contours - the pulsed calls - being the most common and intensively studied vocalization of killer whales.



a) Echolocation Clicks **b)** Whistle **c)** Pulsed Call

Figure 1: *Spectral visualization of the orca vocal repertoire*

Figure 1 visualizes those three different and characteristic orca sound types. Pulsed calls are classified into discrete, variable, and aberrant calls. Sudden and patterned shifts in frequency can be observed, based on the pulse repetition rate, usually between 250 and 2000 Hz [2]. Resident orcas live in stable matrilineal units. Matrilines often traveling together to socialize regularly form subpods and pods [3, 4, 5]. Distinct vocal repertoires of different pods consists of a mixture of unique and shared (between matrilines) discrete call types and are referred to as dialects.



a) A5 N09 **b)** A12 N09 **c)** A24 N09 **d)** A36 N09

Figure 2: *Spectrogram N09 human-labeled call types (red = ref. N09 call, A5 pod; green = N09 calls, same/different pods)*

Each of those pod-specific dialects is made of up to 20 types of discrete calls. The Northern Residents' vocal repertoire of discrete calls consists of more than 40 types [6, 5]. These types have been classified by humans from listening and looking at their signal spectra. The current understanding of orca communication and vocalization is based on the call types which were established by Ford in 1987 [5], including also various whistle types (stereotyped and ultrasonic). Group-specific vocal signals are believed to play an important role in maintaining contact among members or coordinate group activities, especially when groups are dispersed and when visual signals can only be used in short distance communication [2]. Call structure variations can be observed in various shared call types [7]. Obviously there exist huge inter- but also intra-pod signal variations even within one single human-labeled call type. Figure 2 illustrates the wide spectral variety within a given N09 call class, recorded from the A5 pod, compared to N09 calls of the same pod (Figure 2a) and other pods (Figure 2b-d). All these call types were generated based on human perception. By looking at the different versions of calls classified as N09, a valid assumption is that many potentially meaningful details and differences have

been overlooked, due to missing tools in order to compare hundreds and thousands of similar calls. By training machine algorithms to analyze orca calls at a finer level of detail, it is possible to detect either new/updated classes, and/or sub-classes of call types, human-misclassifications, and better understand potentially meaningful differences within the orca communication. The current study explores the possibility of using a multi-step fully unsupervised approach including feature learning based on automatic machine-segmented orca data and call type clustering using deep learning methods to identify machine-generated call type clusters sharing the highest degree of similarity, as a way to automatically generate different call type labels. The proposed method is intended to address issues such as: (1) data annotation (labeled versus unlabeled data), (2) classification based on human perception, (3) human error-proneness, (4) data-driven technique to analyze large (bioacoustic) audio corpora in order to derive totally new insights. By comparing the data-driven call type clusters with human classifications, it demonstrates the prospect of using unsupervised clustering to classify animal vocalizations and to eliminate the blind spots in human perception. The whole unsupervised pipeline was evaluated by using a human-labeled call type dataset (section 4). Moreover, we compared the unsupervised cluster outputs against our previous supervised classification result in [8].

2. Related Work

Salamon et al. [9] improved species classification of 43 bird species by their flight calls using spherical k-means clustering for feature learning to derive a codebook from the training data, a subsequent feature-encoding, followed by a final supervised SVM classification. Brown et al. [10] used Dynamic Time Warping and k-means to cluster 57 captive killer whale vocalizations resulting in 9 distinct clusters compared against human classified call types. Picot et al. [11] used an algorithm consisting of 4 steps – signal processing, segmentation, pattern recognition, and clustering – to unsupervised classify intonations of sound units of the same humpback whale song. Rickwood and Taylor [12] utilized a fully unsupervised approach containing signal identification (power-spectrogram), feature extraction (time-varying power spectra), and unsupervised clustering (vector quantization plus HMMs) to classify humpback song units. Pace et al. [13] classified humpback subunits via segmentation (energy detector) and clustering on MFCCs. To the best of our knowledge, there is no study utilizing deep learning in a fully unsupervised approach including feature learning on machine-labeled data and clustering in order to derive/identify machine-learned orca communication patterns/vocalizations.

3. Methodology

Convolutional Neural Network (CNN) – CNN is an end-to-end deep learning technique in order to efficiently handle, process, and compress the complexity of 2-D input data (e.g. spectrograms) [14]. CNNs implement the traditional principle of pattern recognition – feature learning done by convolutional layers and classification handled via fully connected layers [14]. Convolutional layers are characterized by (1) local receptive fields, (2) weight sharing, and (3) sub-sampling (pooling) [14]. Hidden units (features), stored in one feature map (channel), are generated via sliding a convolutional kernel (k×k constant shared-weight matrix) at a given hop size (stride) over the whole input shape, while extracting at each of the kernels local receptive fields one feature value (linear operation) [14]. In addition to the core concepts of this network architecture – convolution and pooling – CNNs embed activation layers (e.g. Rectified Linear Unit [15] layer) and normalization layers (e.g. batch normalization [16] layer). The type of activation and normalization layers, as well as the sequential layer ordering (convolutional, normalization, activation, and pooling layers), depends on the type of application and data.

Residual Networks (ResNet) – In order to build and train deeper neural network models He et al. [17] invented a resdiual learning framework to counteract the degradation problem. Increasing the network depth leads to an accuracy decrease after the saturation region due to higher training errors compared to shallower counterparts [17]. A network architecture which is not directly learning an underlying mapping $H(x)$ with respect to a given input x but rather optimizing the residual mapping $F(x) = H(x) - x$ is called residual network (ResNet) [17]. For a more detailed information see [17]. In this study we utilized the ResNet18 architecture [17].

Autoencoder – In order to learn adequate and robust feature representations we trained an undercomplete autoencoder to derive a compressed feature representation based on the original input data. An input sample x is encoded via an encoder function e to a hidden representation $h = e(x)$, whereas a decoder function d maps the latent code h to its reconstruction/output $r = d(h)$ [18]. Thus, an autoencoder acts as a copy operation of the input to its output characterized by $x = d(e(x))$, while using several constraints within the compression/decompression process, e.g. dimensionality reduction to learn a compressed embedding h while trying to minimize the loss $L(x, d(e(x)))$ referring to dissimilarity [18].

Clustering – In this work we used spectral clustering with a radial basis function to create the affinity matrix. For a more detailed information we refer to [19].

4. Data Basis

Orca Segmented Data (OSD) – Our ResNet18-based segmenter [8] was used to segment audio tapes from the Orchive [20, 21]. The Orchive comprises ≈20,000 h of orca underwater recordings, captured via stationary hydrophones in northern British Columbia (Hanson Island) over 25 years (1985-2010). The data is available upon request only. Orca sound events out of 238 (≈192 h) noise-heavy, unlabeled, and randomly chosen Orchive tapes, have been automatically segmented by our supervised trained segmenter. The machine-generated OSD dataset includes 19,211 orca segments summing up to 34.47 h. The OSD corpus was split into a training (13,443 samples, 70.0 %), validation (2,902 samples, 15.1 %), and test set (2,866 samples, 14.9 %), which was used for fully unsupervised training of the autoencoder (Figure 3). We ensured that none of the segmented Orchive tapes, used for feature learning, were part of the call type data corpus.

Call Type Data – According to our previous work [8] we used the same call type data corpus consisting of three distinct catalogs: (1) Orcalab call type catalog (CCS), (2) Ness call type catalog (CCN), and (3) an extension catalog (EXT). The CCS dataset includes 33 N01, 10 N02, 21 N04, 14 N05, 18 N07, 26 N09, and 16 N12 orca call types summing up to 138 call types distributed over 7 classes [8]. The CCN corpus contains 36 N01, 56 N03, 60 N04, 31 N07, 70 N09, and 33 N47 orca call types summarized to 286 shared over 6 classes [8]. Both catalogs lead to 9 various call types and 424 samples. The EXT data pool includes 30 echolocation clicks, 30 whistles, and 30 noise files summing up to 90 samples over 3 classes [8]. EXT was added to get closer to a real-world scenario. In total, our entire call type corpus comprises 514 orca sound events, distributed over 12 various classes. To ensure comparability this study used the same train (363 samples, 70.6 %), validation (72 samples, 14.0 %), and test (79 samples, 15.4 %) split of the overall call type dataset as in [8]. Despite the unbalanced label distribution within the distinct call type catalogs we ensured that every call type is present in every data partition.

5. Network Models

Segmenter – In our previous work [8] we described a supervised trained ResNet18-based orca/noise segmenter achieving a test set accuracy of 95.0 % together with ≈94.0 % true postive rate and ≈4.0 % false positive rate. In order to avoid losing too much resolution at the early stages we removed the 3×3 (stride 2) max pooling layer from the first residual layer [8]. The segmenter uses the same architecture as the residual encoder path of our autoencoder visualized in Figure 3, combined with a subsequent 512-D latent layer mapping the global-average-pooled multi-channel features to the orca/noise output layer.

Call Type Classifier – In order to compare unsupervised call type clustering versus supervised call type classification we used our call type classifier of [8], which achieved a mean test accuracy of 87.0 % on the 79 samples of the call type data test set. Our call type classifier utilized the same, slightly modified ResNet18 architecture than our segmenter [8], with the only difference of mapping the latent layer output to a 12-D output layer overcoming our 12-class problem.

Convolutional Undercomplete Autoencoder – In this study we used a ResNet18-based convolutional undercomplete autoencoder visualized in Figure 3. As bottleneck layer we applied 2 convolutional layers using a 1×1 kernel (stride 1) for compressing and decompressing 512×16×8 features of the last residual layer to 4×16×8 features and back. Moreover, the decoder path slightly differs from the encoder path regarding the penultimate residual layer dimensionality. For upsampling we used transposed convolutions. However, transposed convolutions with a stride of 2 lead to potential reconstruction artifacts, which can not be regularized after the last layer. Thats why the original input dimensionality of 256×128 was already reached within the second to last residual layer. To correct as many artifacts as possible we used a transposed convolution with a stride of 1 in the last residual layer (Figure 3).

6. Data Preprocessing and Training

Data Preprocessing – All our proposed models (section 5) utilized the same data preprocessing pipeline. An audio file was converted to a mono signal and resampled at 44.1 kHz [8]. A subsequent short-time Fourier transform (STFT), using a window length of 4,096 (≈ 100 ms) and hop-size of 441 (≈ 10 ms) samples, was processed and the output transformed to a power spectrogram converted to decibel scale. Now various sequential ordered augmentation techniques, all using a uniform distribution [8] were performed: (1) intensity (factor -6 – +3 dB), (2) pitch (factor 0.5 – 1.5), and (3) time (factor 0.5 – 2.0) augmentation [8], followed by a linear frequency compression (fmin = 500 Hz, fmax = 10 kHz) resulting in 256 fequency bins. Moreover, pitch and time augmented characteristic noise files were added to the spectrogram leading to a SNR between -3 and +12 dB [8]. Noise augmentation was only processed during training the segmenter (noise robust) and classifier (limited data) but not the autoencoder, in order to not overrepresent noise. During training each spectral input file was randomly augmented within each epoch. The resulting training spectrogram was normalized via dB-normalization (-100 – +20 dB). Finally a 1.28 s randomly chosen segment of the normalized spectrogram (zero-padding if too short) was extracted resulting in a trainable clip of 256×128 input size [8].

Training – All our models were implemented and trained in PyTorch [22]. They all use an Adam optimizer with an initial learning rate of 10^{-5}, $\beta_1 = 0.5$, $\beta_2 = 0.999$, and learning rate decay of 0.5 after 4 epochs without any improvements on the validation set (accuracy for segmentation/classification, loss for feature learning). The segmenter (cross-entropy loss) was trained on the dataset in [8], the autoencoder (mean squared error (MSE) loss) on the OSD corpus, and the call type classifier (cross-entropy loss) on the call type training set (section 4). The OSD corpus, used for feature learning does not include any tapes from the CCS, CCN, and EXT dataset. For segmentation and feature learning a batch size of 32 was utilized, whereas for call type classification we used a batch size of 4 [8]. The training process for all our models was canceled after 10 epochs without any progress on the validation set [8].

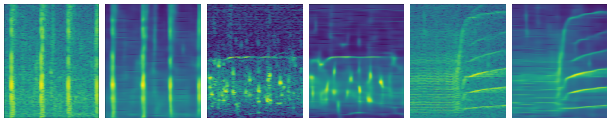
7. Experiments

In our first experiment we performed and evaluated our two-stage fully unsupervised pipeline. Initially the automatic pre-segmented OSD dataset was utilized to train our autoencoder (Figure 3) in a fully unsupervised way, in order to learn meaningful and compressed orca feature representations. The second step was a fully unsupervised spectral clustering [19] of the entire call type dataset (section 4). Therefore, we used the 4×16×8 (512-D) learned feature representations of the autoencoders bottleneck layer (Figure 3) as cluster input. In this experiment we clustered all 514 orca sound samples from the call type data corpus. The number of clusters were calculated via processing the gap statistic [23]. The computation of gap statistics and downstream clustering was done iteratively and recursively for each cluster until the gap statistic of a cluster no longer exceeded a threshold of ≥ 0.5 or the amount of samples in a cluster was less than the call type dataset size (514 samples) divided by the number of clusters from the first clustering run. As a result we ended up with a total number of 29 clusters. In a second experiment we examined the 29 cluster outputs referring to potential call type sub-classes and human-misclassifications for all 514 human-labeled orca sound samples. In our last experiment we only clustered the 79 test samples of our call type test set and forced the cluster algorithm to 12 output clusters, in order to compare it with our previous supervised classification result [8] by visualizing both confusion matrices.

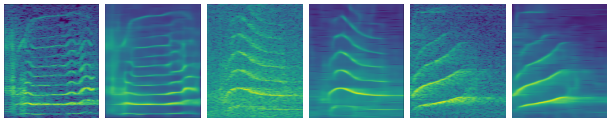
8. Results and Discussion

8.1. Visualization Autoencoder Reconstructions

For a feasibility analysis of such a fully unsupervised pipeline, it is imperative to have an upstream robust orca sound segmentation, an adequate downstream feature learning and a subsequent clustering. Figure 4 shows the spectral reconstruction results of the autoencoder, fully unsupervised trained on the machine-segmented output data (OSD). The reconstruction samples (Figure 4) are examples of the call type dataset and were consequently not part of the autoencoder training. The autoencoder was trained on pre-segmented orca signals and thus reconstructs/reflects the orca data much better than the noise.



echo (orig. vs. rec.) whistle (orig. vs. rec.) N09 (orig. vs. rec.)



N01 (orig. vs. rec.) N04 (orig. vs. rec.) N47 (orig. vs. rec.)

Figure 4: *Original and reconstructed spectrograms of various orca sounds using the ResNet18 undercomplete autoencoder.*

Figure 4 shows that the autoencoder learns the different orca signal variations (overlaying harmonics, clicks, etc.) in de-

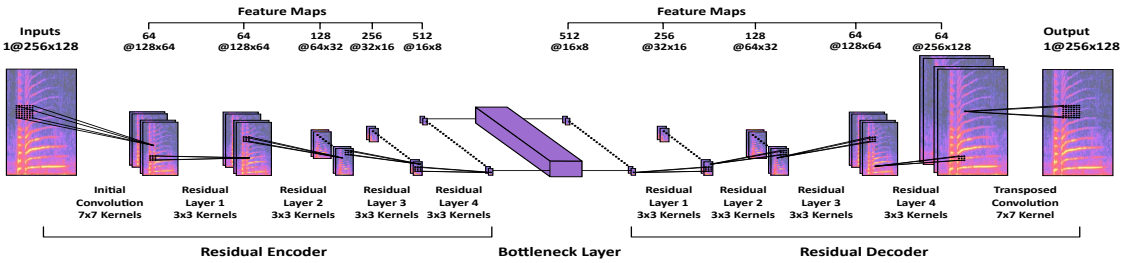
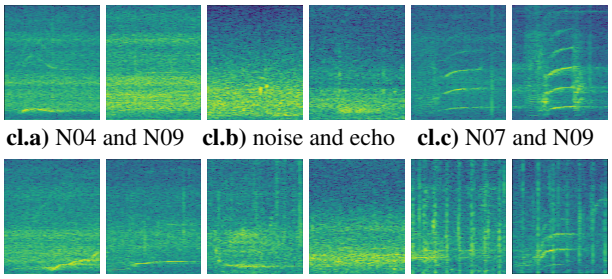


Figure 3: *Network architecture of the ResNet18 undercomplete autoencoder.*

tail, rather than reconstructing the noise parts (blurred, as a result of the standard mean squared error (MSE) loss function [24]), which is a significant indication of generating automatic, very robust orca segmentation results, leading to highly valuable compressed feature representations as input for the spectral clustering.

8.2. Human Misclassifications and Call Type Sub-Classes

The cluster output was verified referring to potential human misclassifications and call type sub-classes. Figure 5 visualizes 6 various clusters (cl.a–cl.f), each describing two samples of the same cluster showing very similar spectral shapes but different human classifications. Moreover, cluster cl.f illustrates a common problem of having a certain call type together with an overlaying very prominent echolocation.

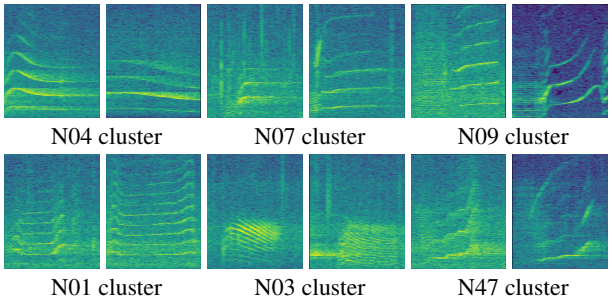


cl.a) N04 and N09 **cl.b)** noise and echo **cl.c)** N07 and N09

cl.d) N47 and N09 **cl.e)** N03 and noise **cl.f)** el and N07

Figure 5: *Cluster outputs containing potential machine-identified human misclassifications for different orca sounds.*

This phenomenon leads to an uncertainty regarding a proper classification of the relevant/interested orca sound type. Those machine-identified and visually comprehensible examples could be an indicator for misclassifications and/or general wrong interpretation of certain call types (N07 vs. N09).



N04 cluster N07 cluster N09 cluster

N01 cluster N03 cluster N47 cluster

Figure 6: *Cluster outputs visualizing potential machine-identified call type sub-classes.*

Moreover, we clustered single human-labeled call type classes (e.g. N04) to identify potential sub-classes. Figure 6 visualizes potential sub-classes of N04, N07, N09, N01, N03, and N47. The spectral variety of identical-labeled call types in human-defined call type classes is clearly observed, even though we can only show very few examples due to lack of space.

8.3. Supervised Classification vs. Unsupervised Clustering

For the comparison of supervised classifcation and unsupervised clustering, we determined the number of clusters to be 12 and exclusively clustered the 79 orca samples of the call type test set (section 4). Thus, it allows us to compare the cluster output with the confusion matrix of our supervised classification in [8]. Despite the suboptimal cluster number, the non-homogeneous distribution of the test labels, possible human misclassifications, and our fully unsupervised pipeline an accuracy of ≈60.0 % was achieved for the 12-class problem.

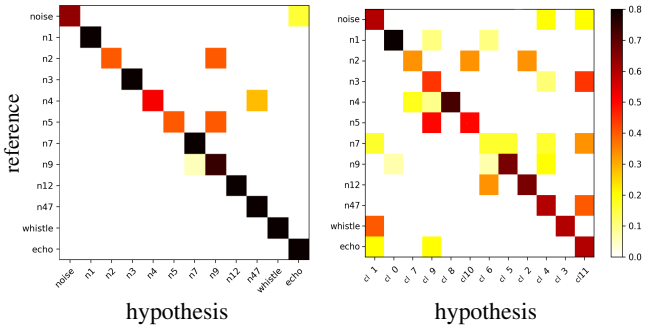


Figure 7: *Superv. Classification [8] vs. Unsuperv. Clustering*

9. Conclusion and Future Work

Our proposed fully unsupervised pipeline, based on machine-segmented orca data, has proven particularly useful for orca call type identification due to the following reasons: (1) no labeled data required (2) less susceptibility to human errors (misclassifications) (3) robust analysis of large datasets (4) human perception eliminated (5) accuracy of ≈60.0 % on a 12-class problem (6) deep analysis enables derivation of new, previously unknown (sub-)call types. In our future work we will use this pipeline to process the entire Orchive (20,000 h) to derive/identify highly valuable orca call type data and new, unseen (sub-)call types, by analyzing data over 25 years in order to facilitate totally new insights and possibilities in animal research.

10. Acknowledgements

The authors would like to thank Helena Symonds and Paul Spong from OrcaLab, and Steven Ness, formerly UVIC, for giving us permission to use the raw data and annotations from the orcalab.org, and the Paul G. Allen Frontiers Group for their initial grant for the pilot research. Moreover, the authors would like to thank Michael Weber for designing the autoencoder image.

11. References

[1] O. A. Filatova, F. I. Samarra, V. B. Deecke, J. K. Ford, P. J. Miller, and H. Yurk, "Cultural evolution of killer whale calls: background, mechanisms and consequences," *Behaviour*, vol. 152, pp. 2001–2038, 2015.

[2] J. K. B. Ford, "Acoustic behaviour of resident killer whales (Orcinus orca) off Vancouver Island, British Columbia," *Canadian Journal of Zoology*, vol. 67, pp. 727–745, January 1989.

[3] T. Ivkovich, O. Filatova, A. Burdin, H. Sato, and E. Hoyt, "The social organization of residenttype killer whales (Orcinus orca) in Avacha Gulf, Northwest Pacific, as revealed through association patterns and acoustic similarity," *Mammalian Biology*, vol. 75, p. 198210, May 2010.

[4] J. Ford, G. Ellis, and K. Balcomb, in *Killer whales: the natural history and genealogy of Orcinus orca in British Columbia and Washington*, 2000.

[5] J. K. B. Ford, "A catalogue of underwater calls produced by killer whales (Orcinus orca) in British Columbia," *Canadian Data Report of Fisheries and Aquatic Science*, no. 633, p. 165, January 1987.

[6] ——, "Vocal traditions among resident killer whales (Orcinus orca) in coastal waters of British Columbia," *Canadian Journal of Zoology*, vol. 69, pp. 1454–1483, June 1991.

[7] P. J. O. Miller, "Diversity in sound pressure levels and estimated active space of resident killer whale vocalizations," *Journal of Comparative Physiology*, 2006.

[8] H. Schröter, E. Nöth, A. Maier, R. Cheng, V. Barth, and C. Bergler, "Segmentation, classification, and visualization of orca calls using deep learning," in *International Conference on Acoustics, Speech, and Signal Processing, Proceedings (ICASSP)*, May 2019 - to appear.

[9] J. Salamon, J. P. Bello, A. Farnsworth, M. Robbins, S. Keen, H. Klinck, and S. Kelling, "Towards the automatic classification of avian flight calls for bioacoustic monitoring," *PLOS ONE*, vol. 11, pp. 1–26, November 2016.

[10] J. Brown, A. Hodgins-Davis, and P. Miller, "Classification of vocalizations of killer whales using dynamic time warping," *JASA Express Letters*, vol. 119, no. 3, pp. 617–628, March 2006.

[11] G. Picot, O. Adam, M. Bergounioux, H. Glotin, and F.-X. Mayer, "Automatic prosodic clustering of humpback whales song," in *New Trends for Environmental Monitoring Using Passive Systems*, November 2008.

[12] P. Rickwood and A. Taylor, "Methods for automatically analyzing humpback song units," *The Journal of the Acoustical Society of America*, vol. 123, pp. 1763–1772, January 2008. [Online]. Available: https://asa.scitation.org/doi/pdf/10.1121/1.2836748?class=pdf

[13] F. Pace, F. Benard, H. Glotin, O. Adam, and P. White, "Subunit definition and analysis for humpback whale call classification," *Applied Acoustics*, vol. 71, pp. 1107–1112, November 2010.

[14] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, vol. 86, November 1998, pp. 2278–2324.

[15] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, June 2010, pp. 807–814.

[16] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, vol. 37, July 2015, pp. 448–456.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[18] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org. [Online]. Available: https://www.deeplearningbook.org/

[19] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in neural information processing systems*, 2002, pp. 849–856.

[20] ORCALAB, "Orcalab - a whale research station on Hanson Island," http://orcalab.org (September 2018). [Online]. Available: http://orcalab.org/

[21] S. Ness, "The Orchive : A system for semi-automatic annotation and analysis of a large collection of bioacoustic recordings," Ph.D. dissertation, 2013.

[22] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *NIPS 2017 Workshop*, October 2017.

[23] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 2, pp. 411–423, 2001.

[24] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," *CoRR*, vol. abs/1511.05440, 2016.