Analyzing Big Environmental Audio with Frequency Preserving Autoencoders

Benjamin Rowe*, Philip Eichinski†, Jinglan Zhang‡, Paul Roe§
Science and Engineering Faculty,
Queensland University of Technology
Brisbane Australia
Email: *benjamin.rowe@hdr.qut.edu.au, †philip.eichinski@qut.edu.au,
‡jinglan.zhang@qut.edu.au, §p.roe@qut.edu.au

***Abstract*—Continuous audio recordings are playing an ever more important role in conservation and biodiversity monitoring, however, listening to these recordings is often infeasible, as they can be thousands of hours long. Automating analysis using machine learning is in high demand. However, these algorithms require a feature representation. Several methods for generating feature representations for these data have been developed, using techniques such as domain-specific features and deep learning. However, domain-specific features are unlikely to be an ideal representation of the data and deep learning methods often require extensively labeled data.**

In this paper, we propose a method for generating a frequency-preserving autoencoder-based feature representation for unlabeled ecological audio. We evaluate multiple frequency-preserving autoencoder-based feature representations using a hierarchical clustering sample task. We compare this to a basic autoencoder feature representation, MFCC, and spectral acoustic indices. Experimental results show that some of these non-square autoencoder architectures compare well to these existing feature representations.

This novel method for generating a feature representation for unlabeled ecological audio will offer a fast, general way for ecologists to generate a feature representation of their audio, which does not require extensively labeled data.

***Index Terms*—Machine Learning, Deep Learning, Autoencoder, Ecoacoustics**

I. Introduction

Conventional survey methods, such as point counts, typically involve experienced surveyors identifying species in the field. [1]. The cost associated with these conventional survey methods and the falling cost of digital recording equipment has resulted in continuous audio recordings playing an increasingly important role in environmental conservation. However, these recordings can often be thousands of hours long, making it infeasible to listen to them in their entirety. This has lead to many different techniques being employed to analyze these data, each with its own advantages and drawbacks.

We aim to design a technique for generating a feature representation for ecological audio which can be used for fast, general and fairly accurate analysis.

It has previously been shown that basic autoencoder-based representations offer comparable performance to spectral acoustic indices with a reduced computational cost [2]. The nature of spectrograms of vocalizing fauna means that the feature representation should be sensitive to translations in the frequency direction but not in the time direction. We hypothesize an autoencoder-based architecture utilizing CNNs that discard more information in the time direction and preserve more information in the frequency direction (frequency-preserving autoencoders) can be more accurate than a basic autoencoder when working with bird audio. This approach is not been examined before.

The contribution of this work will be 1) a new method for producing a feature representation from unlabeled ecological audio which preserves frequency information, and can be used as part of a framework for analyzing and exploring ecological audio using machine learning or visualization, and

2) a preliminary understanding of how to set the hyperparameters of these networks to produce a more robust feature representation.

II. Related Work

A. Soundscape Ecology

The ability of ecosystems to function is decreasing in response to the effect of humans on global biodiversity [3]. This growing pressure from human activities and climate change has increased the importance of biodiversity monitoring in recent years.

The new field of soundscape ecology [4] is described by Pijanowski et. al. as "...all sounds, those of biophony, geophony, and anthrophony, emanating from a given landscape to create unique acoustical patterns across a variety of spatial and temporal scales" [5]. The cost associated with continuous audio recording was previously seen as a significant issue for soundscape ecology [4]. However, due to the falling cost of audio recording equipment [1], continuous audio recordings are playing an ever more important role in conservation and biodiversity monitoring. These recordings are often infeasible to listen to due to their length. As a result, soundscape ecologists often employ a variety of machine learning and visualization tools to assist with analyzing these data.

Ecoacoustic data can vary with location and time [6], and many species produce sounds that have not been documented due to the species being uncommon or other factors [7]. As such, many techniques utilise supervised or semi-supervised machine learning techniques that require annotated training data from each target site and species, or utilise human-selected domain-specific features such as "acoustic indices" [8] [9] [10] [11], which often require careful selection and fine-tuning using the knowledge of a domain expert [12]. In the case of acoustic indices, they are known to behave poorly at small timescales [13].

B. Auto-Encoders

Autoencoders are a type of neural network that aim to produce a similar output to their input, with constrained hidden layers of progressively smaller size. They consist of an encoder network that reduces the dimensionality of the input data using several hidden layers. The output of the encoder network (the center hidden layer) is then used as the input of the decoder network. The activation map of the center hidden layer can then be used as a feature vector for visualization or other machine learning techniques [14].

Auto-encoders were initially introduced as a method for dimensionality reduction [15], and when using linear activation functions can yield equivalent equivalent results to those obtained using principal component analysis (PCA) [16]. However auto-encoders with non-linear activation functions are not equivalent to PCA [17] and have been shown to produce better results when used as a dimensionality reduction step prior to classifying data with a non-linear decision boundary [18].

There are common variations of auto-encoder, including "Denoising Auto-encoders" which aim to reconstruct a reduced noise output [19], "Variational Auto-encoders" which aim to produce a variation of the input as the output and "stacked Auto-encoders", which use the encoded representation as the input and output of a nested auto-encoder [20]. These variations of basic auto-encoders have been found to produce more robust features [21], and have additional uses (such as denoising) which make them more versatile.

C. Non-Square Convolutional Neural Networks

In typical CNN autoencoders, hyperparameters for the shapes and strides for padding and kernels are the same in the vertical and horizontal direction, resulting in equal dimensionality reduction in each direction. However, spectrogram horizontal and vertical directions mean quite different things from each other, so it is not obvious that this approach would be best. Despite this, CNN autoencoders with non-square hyperparameters has not yet been documented for use with ecological audio.This technique has, however, seen use with autoencoders and other CNN-based networks when working with non-audio data.

Much of the literature exploring non-square (also known as asymmetric) kernels in CNNs focuses on

attempts to produce networks that are equivalent to those using square kernels while yielding a decrease in computation time [22]. This is performed with varying levels of effectiveness [23] [24] [25].

In "ACNet: Strengthening the Kernel Skeletons for Powerful CNN via Asymmetric Convolution Blocks", Ding, et. al. propose a network architecture based on "Asymmetric Convolution Blocks" (ACBs), which they call "ACNet" [22]. ACBs use a kernel shape that is equivalent to summing a traditional 3x3 kernel, with a 3x1 and a 1x3 kernel. They find that ACNet outperforms an off-the-shelf counterpart for two image classification tasks. Karström and Landgren explore the use of CNNs with non-square kernels in "Relative Pose Regression using Non-Square CNN Kernels" [26]. They explore CNNs that use translating (T-CNN), rotating (R-CNN), and scaling (S-CNN) kernels, and aim to determine if these three alternative kernel shapes offer an improvement when detecting tralslation along one axis, rotation, and scaling, compared with "normal" CNNs. They found that for their chosen tasks, their T-CNN performed comparably to their "normal" CNN, R-CNN performed better than their "normal" CNN and their S-CNN performed worse than their "normal" CNN.

The use of auto-encoders with non-square layers is currently uncommon. However, papers utilizing such auto-encoders usually do so to make use of the speed improvements that are offered by utilizing alternating 1xn and nx1 convolutional layers. Researchers have utilized autoencoder-inspired networks for semantic image segmentation, including "ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation", "U-Net: Convolutional Networks for Biomedical Image Segmentation", and "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation" [27] [28] [29].

D. Auto-Encoders and Ecological Audio

Basic autoencoders have been previously explored as a method for generating features for ecological audio [**?**]. It is found that at a 1-second timescale and with minimal training, the features extracted by auto-encoders perform comparably to spectral acoustic indices, but autoencoders were capable of extracting features 2 orders of magnitude faster. As such, this method could be judged to be fast when compared to its alternatives. However, some spectral indices are known to "misbehave" at small timescales [13] thus the accuracy is low, so the accuracy of basic autoencoders should be considered "low".

III. Method

A. Dataset

In order to train with a variety of data, the chosen dataset consisted of audio recorded at 3 sites. The locations of these sites are illustrated in Figure 1.

a) The SERF dataset: consists of approximately 20 days of continuous audio recordings collected from 4 audio recorders at the Samford Ecological Research Facility (SERF), located in Samford, Moreton Bay Region, South East Queensland, Australia [30]. SERF is located northwest of Brisbane, in the Samford Valley. "70% of the property is covered with vegetation providing a refuge to native plants and animals that are under increasing pressure from urbanization" [31]. The audio was sampled at 22050Hz in 16-bit stereo, which was compressed using MP3 compression to reduce file size for storage on 32GB SD card (considered high capacity at the time of recording) [32], and later converted to Mono.

b) The Gympie National Park dataset: consists of approximately 1 year of continuous audio data, followed by approximately 1 year of sparse audio recording collected from 1 audio recorder at Gympie National Park, located in North Deep Creek, Gympie Region, South East Queensland, Australia [33]. The audio was sampled at 22050Hz in 16-bit stereo and stored uncompressed. However, for this project, the audio was compressed with mp3 to remain consistent.

c) The Woondum National Park dataset: consists of approximately 1 year of continuous audio data, followed by approximately 1 year of sparse audio recording collected from 1 audio recorder at Woondum National Park, located in Mothar Mountain, Gympie Region, South East Queensland, Australia [34]. The audio was sampled at 22050Hz

in 16-bit stereo and stored uncompressed. However, for this project, the audio was compressed with mp3 to remain consistent.

These datasets were chosen as it has been highly annotated and used in previous studies. The extensive annotation of the dataset allowed the feature representation produced by the auto-encoder to be easily evaluated and to be trained on a higher proportion of non-silent audio clips.

Fig. 1. The sites in Australia

B. Data Pre-processing

The dataset was first segmented into non-overlapping 1-second audio clips. All audio clips collected from a recorder located in the southeast of SERF were first set aside to be used as testing data. Then from the three remaining recorders on the property, approximately 320,000 1-second audio clips were selected from 30-second segments known to contain at least 1 annotation. These 320,000 seconds were then used as training data. These audio clips were then converted into (1-second-long) grayscale audio spectrograms using fast fourier transform (FFT), with a window size of 256 (approximately 0.01 seconds), overlap of 128, and the hanning windowing function, using python and matplotlib. Each output spectrogram was then resized using bilinear interpolation to 128x128 pixels. Using a power-of-2 input size significantly simplified further processing, allowing each convolutional layer to be half the size of its predecessor.

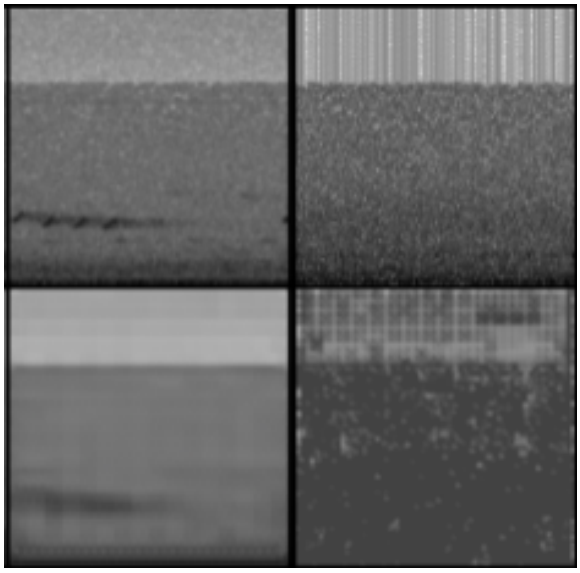

Fig. 2. Example of checkerboarding. (top) example input spectrograms (bottom-left) example using implicit unpooling (bottom right) example using max-unpooling in which checkerboarding can be seen

C. Architecture

We will utilize CNNs with non-square kernels to discard time information and preserve frequency information. As there is very little literature exploring autoencoders with non-square kernels and strides when creating our frequency preserving architecture we elected to create several different encoder and decoder networks. We allowed networks consisting of asymmetric encoder/decoder pairings. This allowed us to test several different methods of encoding and decoding together, to allow us to get a better understanding of which methods work best for encoding and decoding. All compatible encoder to decoder combinations were tested. The details of each encoder and decoder network can be seen in Table I, and the possible combinations can be seen in Table II. An example network can be seen in Figure 3.

For all networks, an encoder input size of 128x128x3 (128 pixels wide, 128 pixels high, and 3 repeated color channels) and decoder output size of 128x128x3 were used. On every network, a rec-

tified linear unit (ReLU) based activation function was used, to help mitigate the vanishing/exploding gradient problem [35].

D. Training

Each encoder/decoder network pairing was trained for 40 epochs using 10 epochs each at decreasing learning rates (1e-3 $\rightarrow$ 1e-4 $\rightarrow$ 1e-5 $\rightarrow$ 1e-6), with the network being saved every 10 epochs. The networks were trained using pre-processed data containing at least 1 annotation, from 3 of the 4 recorders (which were recorded during the chosen time period) at the SERF site.

The loss was given by the mean of the squared difference between each color channel of each pixel in the input and its counterpart in the output. The "adam" optimization algorithm was used. Every compatible pairing (Table II) of encoder and decoder (Table I) network was trained.

E. Evaluation

Hierarchical clustering was used as an example task to evaluate the performance of each frequency preserving architecture, compared to basic autoencoders, MFCC, and spectral acoustic indices. The data used for clustering were taken from 1 of the 4 recorders at SERF. To select the data;

1) The 25 most commonly annotated species were found
2) For each of the 25 species, 100 annotated 1-second recordings were downloaded
3) From these 25 x 100 1-second recordings, calls with a lot of background noise or a lot of calls from species other than the target were removed
4) Where possible, the 25 most similar remaining calls were then saved, and the rest of the recordings for the species were discarded
    a) If more than 25 calls were acceptable, 25 calls were chosen randomly from the acceptable calls
    b) If there were less than 25 similar calls and more than 15, all available calls were used
    c) If there were less than 15 calls the species was removed

This resulted in 554 total 1-second long recordings, from 23 species. The chosen species were:

1) Burhinus grallarius, 2) Chalcites lucidus, 3) Chenonetta jubata, 4) Coracina novaehollandiae, 5) Corvus orru, 6) Cracticus nigrogularis, 7) Eopsaltria australis, 8) Eudynamys orientalis, 9) Geopelia striata, 10) Lichenostomus chrysops, 11) Meliphaga lewinii, 12) Melithreptus albogularis, 13) Myiagra rubecula, 14) Myzomela sanguinolenta, 15) Oriolus sagittatus, 16) Pardalotus striatus, 17) Philemon citreogularis, 18) Rhipidura albiscapa, 19) Sericornis frontalis, 20) Todiramphus sanctus, 21) Trichoglossus haematodus, 22) Vanellus miles, 23) Zosterops lateralis

To evaluate the feature representation, features were generated from the chosen 1-second audio clips. These features were then used as input data for the sample hierarchical clustering task. Hierarchical clustering was chosen as it will allow us to determine how well the feature representation can be used to differentiate between classes. For the hierarchical clustering, the process the number of clusters was set to 10, the linkage to "ward", and the affinity to "euclidean".

To ensure that our evaluation is rigorous, a process similar to "bootstrapping" was used. This consisted of running the sample task 20 times, randomly choosing 10 calls from 10 chosen species each time. The average, standard deviation, and variation across the 20 repetitions were then taken for our chosen clustering evaluation metrics.

B-cubed precision, B-cubed recall, B-cubed fscore, and purity were chosen as our cluster evaluation metrics. They were chosen as they commonly used metrics.

IV. Results

The results of evaluating the hierarchical clustering using B-cubed and purity can be seen in Table II. Networks producing standard deviations and variances of 0 were unable to reproduce a spectrogram and were assumed to be nonfunctional. As such, they were not included in the analysis. Some networks performed worse with superfluous training. Due to this, the best-performing version of each network was used (as saved every 10 epochs).

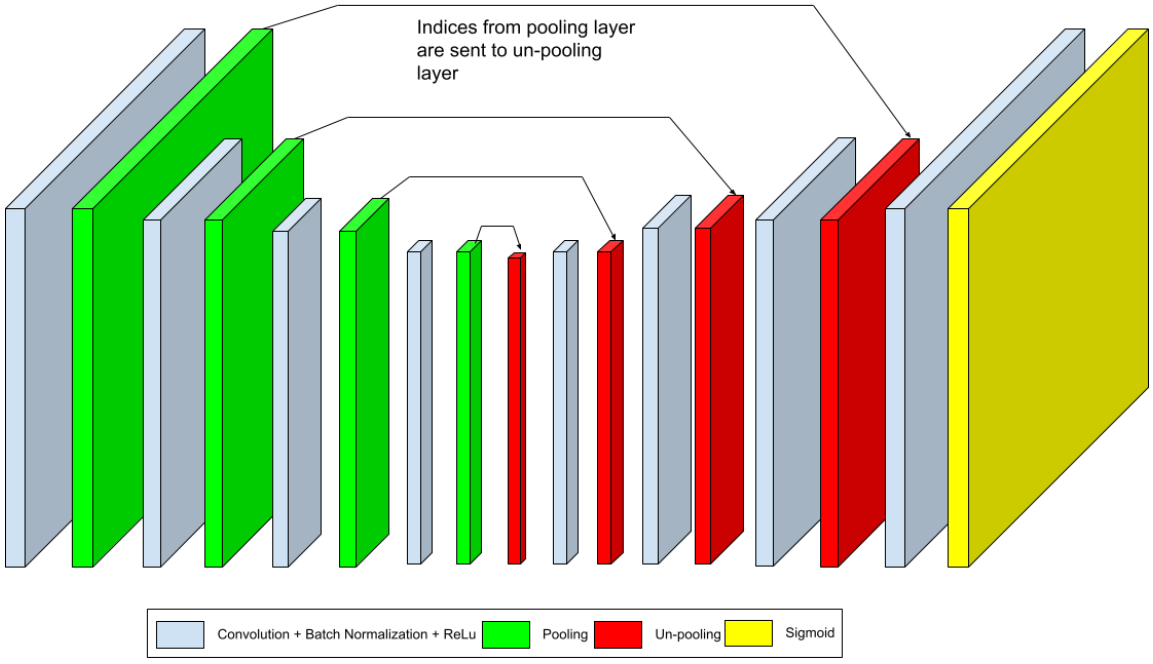

Fig. 3. The structure of an auto-encoder

Of the non-square networks tested, 5 networks achieved an average fscore the same or better than the best square network tested and 2 networks achieved an average purity the same or better than the best square network tested. Two of the networks that achieved a better average purity also achieved a better average fscore. These two networks that performed the best are:

1) "Non-Square (Max-pooling)(Batch Normalization)(Reduce x and y)" "Non-Square (Max-pooling)(Batch Normalization)(Reduce x and y)"
2) "Non-Square (Max-pooling)(Batch Normalization)" "Non-Square(Implicit-pooling)"

Both of these networks achieved an average purity higher than spectral acoustic indices, however, they achieved a lower average fScore. Dimensionally reduced plots of the hierarchical clustering for these two networks can be seen in Figure 4 and Figure 5. Visual analysis of these plots suggests that the networks work well for some species (Species 5 in Figure 4 and species 3, 7, and 8 in Figure 5), but are unable to distinguish between a number of species. This appears to be a better result than was achieved than the best "square" network we tested which can be seen in Figure 6. Although this network seems to be somewhat able to differentiate between species, it does not appear to be able to draw clear boundaries between any of these species.

V. Discussion

A. Findings

The use of non-square autoencoders on ecological audio is novel, and as such, there is no baseline for performance. The most similar research conducted by Rowe, et. al. focuses on basic autoencoders. The basic (square) autoencoders used in this research achieved similar performance to those produced by Rowe, et. al [**?**]. They hypothesized that autoencoders using max-unpooling would perform better when trained for more epochs. Our results suggest this is not the case, with most of

Type	Convolution Kernel	Convolution Stride	Convolution Padding	Pooling Kernel	Pooling Stride	Pooling Padding	Representation Size
Square (Implicit-pooling) (Batch Normalization)	4	2	1	n/a	n/a	n/a	384
Square (Max-pooling) (Batch Normalization)	3	1	1	4	2	1	384
Square (Max-pooling) (Batch Normalization) (Large)	3	1	1	4	2	1	1472
Non-Square (Max-pooling)	3	1	1	[1, 4]	[1, 2]	[0, 1]	3072
Non-Square (Max-pooling) (Batch Normalization)	3	1	1	[1, 4]	[1, 2]	[0, 1]	3072
Non-Square (Implicit-pooling)	[1, 4]	[1, 2]	[0, 1]	n/a	n/a	n/a	3072
Non-Square (Implicit-pooling) (Batch Normalization)	[1, 4]	[1, 2]	[0, 1]	n/a	n/a	n/a	3072
Bilinear Interpolation	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Non-Square (Max-pooling) (Batch Normalization) (Small)	3	1	1	[1, 4]	[1, 2]	[0, 1]	1024
Non-Square (Max-pooling) (Batch Normalization) (Reduce x and y)	3	1	1	[1, 4]	[1, 2]	[0, 1]	1536

TABLE I
THE HYPERPARAMETERS OF THE ENCODER AND DECODER NETWORKS TESTED

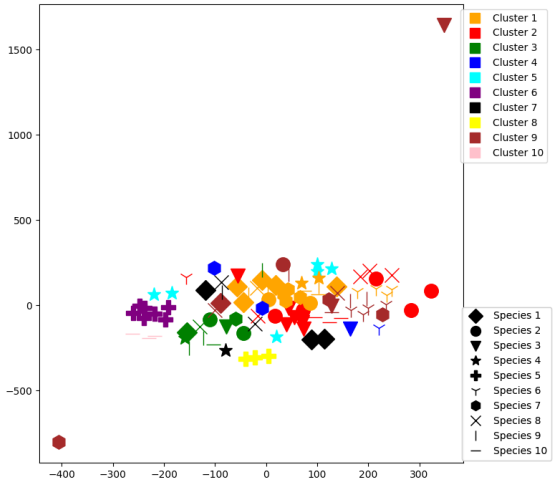

Fig. 4. t-SNE Plot of the "Non-Square (Max-pooling)(Batch Normalization)(Reduce x and y)" "Non-Square (Max-pooling)(Batch Normalization)(Reduce x and y)" network

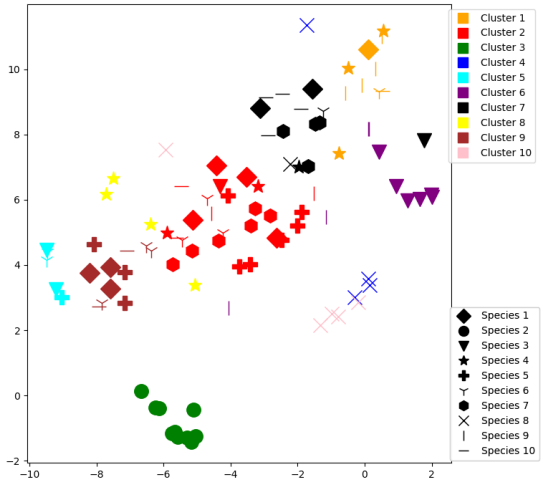

Fig. 5. t-SNE Plot of the "Non-Square (Max-pooling)(Batch Normalization)" "Non-Square(Implicit-pooling)" network

our networks performing the best with our lowest number of epochs. Another similar research by

Encoder	Decoder	Training Epochs	Fscore Average	Purity Average	Fscore Standard Deviation	Purity Standard Deviation	Fscore Variance	Purity Variance
Non-Square (Max-pooling) (Batch Normalization)(Small)	Non-Square (Max-pooling) (Batch Normalization)(Small)	10	0.31	0.19	0	0	0	0
Non-Square (Max-pooling) (Batch Normalization)	Non-Square (Max-pooling)	10	0.27	0.24	0.023	0.020	5.19E-4	4.19E-4
Non-Square (Max-pooling) (Batch Normalization)	Non-Square (Max-pooling) (Batch Normalization)	20	0.3	0.36	3.4E-2	3.8E-2	1.18E-3	1.51E-3
Non-Square (Max-pooling) (Batch Normalization)	Non-Square (Implicit-pooling)	10	0.35	0.41	4.22E-2	4.44E-2	1.79E-3	1.97E-3
Non-Square (Max-pooling) (Batch Normalization)	Non-Square (Implicit-pooling) (Batch Normalization)	10	0.34	0.39	3.7E-2	4.21E-2	1.37E-3	1.77E-3
Non-Square (Max-pooling) (Batch Normalization)	Bilinear Interpolation	20	0.22	0.29	2.58E-2	3.5E-2	6.64E-3	1.22E-2
Non-Square (Max-pooling) (Batch Normalization) (Reduce x and y)	Non-Square (Max-pooling) (Batch Normalization) (Reduce x and y)	10	0.36	0.44	3.8E-2	3.8E-2	1.44E-3	1.44E-3
Non-Square (Max-pooling) (Batch Normalization) (Reduce x and y)	Bilinear Interpolation	10	0.22	0.29	2.75E-2	3.6E-2	7.58E-4	1.3E-3
Non-Square (Max-pooling) (Batch Normalization) (Reduce x and y) (large)	Non-Square (Max-pooling) (Batch Normalization) (Reduce x and y) (large)	10	0.26	0.33	3.8E-2	3.64E-2	1.4E-3	1.33E-3
Non-Square (Max-pooling) (Square Kernel)	Bilinear Interpolation	30	0.21	0.28	2.45E-2	3.48E-2	5.99E-4	1.21E-3
Non-Square (Max-pooling)	Non-Square (Max-pooling)	20	0.26	0.32	4.0E-2	3.88E-2	1.64E-2	1.51E-2
Non-Square (Max-pooling)	Non-Square (Max-pooling) (Batch Normalization)	20	0.26	0.32	3.78E-2	3.97E-2	1.43E-3	1.57E-3
Non-Square (Max-pooling)	Non-Square (Implicit-pooling)	30	0.33	0.4	4.4E-2	5.06E-2	2.12E-3	2.56E-3
Non-Square (Max-pooling)	Non-Square (Implicit-pooling) (Batch Normalization	10	0.28	0.34	3.14E-2	3.11E-2	9.86E-4	9.64E-4
Non-Square (Max-pooling)	Bilinear Interpolation	10	0.31	0.19	0	0	0	0
Non-Square (Implicit-pooling) (Batch Normalization)	Non-Square (Implicit-pooling)	10	0.32	0.37	2.88E-2	3.9E-2	8.3E-4	1.52E-3
Non-Square (Implicit-pooling) (Batch Normalization)	Non-Square (Implicit-pooling) (Batch Normalization)	10	0.33	0.37	4.31E-2	5.0E-2	1.86E-3	2.51E-3
Non-Square (Implicit-pooling) (Batch Normalization)	Bilinear Interpolation	30	0.22	0.29	2.88E-2	3.64E-2	8.32E-4	1.33E-3
Non-Square (Implicit-pooling)	Non-Square (Implicit-pooling)	30	0.29	0.34	2.93E-2	3.97E-2	8.56E-4	1.56E-3
Non-Square (Implicit-pooling)	Non-Square (Implicit-pooling) (Batch Normalization	10	0.31	0.36	4.37E-2	4.0E-2	1.91E-3	1.60E-3
Non-Square (Implicit-pooling)	Bilinear Interpolation	10	0.31	0.19	0	0	0	0
Bilinear Interpolation	N/A	N/A	0.29	0.35	4.09E-2	4.77E-2	1.67E-3	2.28E-3
Square (Implicit-Pooling) (Batch Normalization)	Square (Implicit-Pooling) (Batch Normalization)	10	0.33	0.38	4.5E-2	4.8E-2	2.02E-3	2.31E-3
Square (Max-Pooling) (Batch Normalization)	Square (Max-Pooling) (Batch Normalization)	20	0.23	0.29	3.09E-2	3.2E-2	9.55E-4	1.02E-3
Square (Max-Pooling) (Batch Normalization)	Square (Implicit-Pooling) (Batch Normalization)	10	0.28	0.35	3.66E-2	3.77E-2	1.34E-3	1.42E-3
Square (Max-Pooling) (Batch Normalization)(Large)	Square (Max-Pooling) (Batch Normalization)(Large)	10	0.24	0.31	3.31E-2	4.31E-2	1.09E-3	1.85E-3

TABLE II
ALL FUNCTIONING ENCODER TO DECODER PAIRINGS DERIVED FROM NETWORKS DEFINED IN TABLE I

Dias, et. al, uses varying length spectrograms and does not focus on individual call types directly [36]. As such, it is difficult to compare with this work.

At this 1-second timescale, we found that some networks based on non-square autoencoders outperform square autoencoders, and spectral acoustic indices when performing our hierarchical clustering sample task.

These experimental results show that at a 1-second timescale, some of the tested methods for generating features from environmental audio using non-square autoencoders yield improvements over spectral acoustic indices and basic autoencoders. Although these techniques are outperformed by MFCC they are still feasible for situations in which MFCC is not appropriate, such as for species that

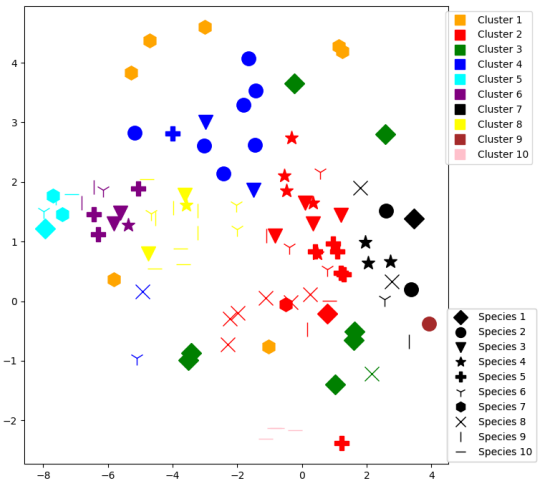

Fig. 6. t-SNE Plot of the best tested square network

may not be represented well by MFCC, or situations in which easy access to frequency information is required. As these autoencoders preserve the location of frequency information in their representation, they may be feasible for use in applications in which having access to frequency information is desirable, such as visualization.

B. Limitations

Our testing with data from a different recorder at the same site suggests that the method could be viable for use as a general approach that does not need to be retrained for each dataset, however, to verify this further testing using data from a different site would be required.

VI. Conclusion

Long duration recordings are playing an ever-more important role in conservation, however, these recordings can be thousands of hours long, making them infeasible to listen to. In this paper, we explored a method for generating a feature representation from ecological audio which can be used with unlabeled ecological audio. We used non-square autoencoders which aim to produce a feature representation that preserves the location of frequency information while discarding some time information. We found that at a 1-second timescale, some such network architectures are able to outperform basic autoencoders and spectral acoustic indices in our sample clustering task when evaluated using b-cubed and purity cluster evaluation metrics.

Non-square autoencoders offer an exciting new alternative for analyzing and exploring ecological audio which does not require extensively annotated training data or expert fine-tuning of parameters. Our next task will be using features generated using these networks for visualization.

References

[1] R. Gibb, E. Browning, P. Glover-Kapfer, and K. E. Jones, "Emerging opportunities and challenges for passive acoustics in ecological assessment and monitoring," *Methods in Ecology and Evolution*, vol. 10, no. 2, pp. 169–185, 2019.

[2] "Acoustic auto-encoders for biodiversity assessment," 2021.

[3] F. S. Chapin III, E. S. Zavaleta, V. T. Eviner, R. L. Naylor, P. M. Vitousek, H. L. Reynolds, D. U. Hooper, S. Lavorel, O. E. Sala, S. E. Hobbie *et al.*, "Consequences of changing biodiversity," *Nature*, vol. 405, no. 6783, p. 234, 2000.

[4] B. C. Pijanowski, A. Farina, S. H. Gage, S. L. Dumyahn, and B. L. Krause, "What is soundscape ecology? an introduction and overview of an emerging new science," *Landscape ecology*, vol. 26, no. 9, pp. 1213–1232, 2011.

[5] B. C. Pijanowski, L. J. Villanueva-Rivera, S. L. Dumyahn, A. Farina, B. L. Krause, B. M. Napoletano, S. H. Gage, and N. Pieretti, "Soundscape Ecology: The Science of Sound in the Landscape," *BioScience*, vol. 61, no. 3, pp. 203–216, 03 2011. [Online]. Available: https://doi.org/10.1525/bio.2011.61.3.6

[6] I. Himawan, M. Towsey, and P. Roe, "3d convolution recurrent neural networks for bird sound detection," in *Proceedings of the 3rd Workshop on Detection and Classification of Acoustic Scenes and Events*, M. Wood, H. Glotin, D. Stowell, and Y. Stylianou, Eds. http://dcase.community/: Detection and Classification of Acoustic Scenes and Events, 2018, pp. 1–4. [Online]. Available: https://eprints.qut.edu.au/122760/

[7] T. Dema, M. Brereton, and P. Roe, "Designing participatory sensing with remote communities to conserve endangered species," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, 2019, p. 664.

[8] S. Gage and A. Axel, "Visualization of temporal change in soundscape power of a michigan lake habitat over a 4-year period," *Ecological Informatics*, vol. 21, pp. 100–109, 2014, cited By 35. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84899989970&doi=10.1016\%2fj.ecoinf.2013.11.004&partnerID=40&md5=42a5970d92851409f985437fd4759cbc

[9] M. Towsey, E. Znidersic, J. Broken-Brow, K. Indraswari, D. M. Watson, Y. Phillips, A. Truskinger, P. Roe, G. Street *et al.*, "Long-duration, false-colour spectrograms for detecting species in large audio data-sets," *Journal of Ecoacoustics*, vol. 2, p. IUSWUI, 2018.

[10] Y. Phillips, M. Towsey, and P. Roe, "Revealing the ecological content of long-duration audio-recordings of the environment through clustering and visualisation," *PLoS ONE*, vol. 13, no. 3, 2018, cited By 0.

[11] H. Gan, J. Zhang, M. Towsey, A. Truskinger, D. Stark, B. van Rensburg, Y. Li, and P. Roe, "Recognition of frog chorusing with acoustic indices and machine learning," in *IEEE 15th International Conference on eScience*. IEEE, 2019.

[12] F. Isensee, P. F. Jaeger, P. M. Full, I. Wolf, S. Engelhardt, and K. H. Maier-Hein, "Automatic cardiac disease assessment on cine-mri via time-series segmentation and domain specific features," in *International workshop on statistical atlases and computational models of the heart*. Springer, 2017, pp. 120–129.

[13] M. W. Towsey, A. M. Truskinger, and P. Roe, "The navigation and visualisation of environmental audio using zooming spectrograms," in *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, 2015, pp. 788–797.

[14] M. Längkvist, L. Karlsson, and A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," *Pattern Recognition Letters*, vol. 42, pp. 11 – 24, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865514000221

[15] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," in *Proceedings of the 28th International Conference on International Conference on Machine Learning*. Omnipress, 2011, pp. 833–840.

[16] P. Baldi and K. Hornik, "Neural networks and principal component analysis: Learning from examples without local minima," *Neural networks*, vol. 2, no. 1, pp. 53–58, 1989.

[17] N. Japkowicz, S. J. Hanson, and M. A. Gluck, "Nonlinear autoassociation is not equivalent to pca," *Neural computation*, vol. 12, no. 3, pp. 531–545, 2000.

[18] N. Japkowicz, C. Myers, M. Gluck *et al.*, "A novelty detection approach to classification," in *IJCAI*, vol. 1, 1995, pp. 518–523.

[19] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, "Speech enhancement based on deep denoising autoencoder." in *Interspeech*, 2013, pp. 436–440.

[20] L. Deng and D. Yu, "Deep learning for signal and information processing," *Microsoft Research Monograph*, 2013.

[21] A. Vafeiadis, D. Kalatzis, K. Votis, D. Giakoumis, D. Tzovaras, L. Chen, and R. Hamzaoui, "Acoustic scene classification: From a hybrid classifier to deep learning," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*, November 2017, pp. 123–127.

[22] X. Ding, Y. Guo, G. Ding, and J. Han, "Acnet: Strengthening the kernel skeletons for powerful cnn via asymmetric convolution blocks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1911–1920.

[23] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[24] J. Jin, A. Dundar, and E. Culurciello, "Flattened convolutional neural networks for feedforward acceleration," *arXiv preprint arXiv:1412.5474*, 2014.

[25] S.-Y. Lo, H.-M. Hang, S.-W. Chan, and J.-J. Lin, "Efficient dense modules of asymmetric convolution for real-time semantic segmentation," in *Proceedings of the ACM Multimedia Asia*, 2019, pp. 1–6.

[26] J. Karström and Ö. Landgren, "Relative pose regression using non-square cnn kernels: Estimation of translation, rotation and scaling between image pairs with custom layers," Master's thesis, Chalmers tekniska högskola / Institutionen för mekanik och maritima vetenskaper, 2020.

[27] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "Enet: A deep neural network architecture for real-time semantic segmentation," *arXiv preprint arXiv:1606.02147*, 2016.

[28] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[29] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[30] J. Wimmer, M. Towsey, B. Planitz, P. Roe, and I. Williamson, "Scaling acoustic data analysis through collaboration and automation," in *2010 IEEE Sixth International Conference on e-Science*, Dec 2010, pp. 308–315.

[31] Qut — samford ecological research facility — serf overview. QUT. [Online]. Available: http://www.serf.qut.edu.au/about/overview/index.jsp

[32] M. Towsey, J. Wimmer, I. Williamson, and P. Roe, "The use of acoustic indices to determine avian species richness in audio-recordings of the environment," *Ecological Informatics*, vol. 21, pp. 110 – 119, 2014, ecological Acoustics. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1574954113001209

[33] Ecosounds — ecosounds — project cooloola — site gympie np. [Online]. Available: https://www.ecosounds.org/projects/1029/sites/1192

[34] Ecosounds — ecosounds — project cooloola — site woondum np. [Online]. Available: https://www.ecosounds.org/projects/1029/sites/1193

[35] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv preprint arXiv:1505.00853*, 2015.

[36] F. F. Dias, H. Pedrini, and R. Minghim, "Soundscape segregation based on visual analysis and discriminating features," *Ecological Informatics*, p. 101184, 2020.