Are You Seeing What I'm Hearing?: A Survey of Audio Feature Engineering Methods for Classification

Nadine Hughes*, Sammy Gagou*, Julia Guzzo*, Aneesh Sallaram*, Joseph Ross*
Department of Computer Science
University of North Carolina at Chapel Hill
Chapel Hill, NC 27514
{nadie, sgagou, jkguzzo, sallaram, ross004}@unc.edu
(*Equal contribution.)

https://github.com/josephross004/560proj

Abstract

This paper explores the results of the study to see how effectively a convolutional neural network (CNN) model can be trained to analyze a spectrogram image of an audio recording and classify the species of animal that made the sound. The dataset was collected from a public GitHub repository [6] and consists of 875 animal sounds and is split into 10 different types (cat, dog, bird, cow, lion, sheep, frog, chicken, donkey, and monkey). Results showed significantly improved processing efficiency and classification accuracy for spectrogram images (75.00%) versus waveforms (39.77%), and a percentile-based compression of spectrograms showed another significant decrease in training time for a comparably small decrease in classification accuracy (60.23%).

1 Introduction and Motivation

Spectrogram classification is useful for a variety of reasons, such as, but not limited to, biodiversity monitoring, animal behavior studies, taxonomic research, and conservation efforts. For example, with more and more species being placed on protected and endangered lists, scientists need a way to track the location and population levels of these species accurately. While we can record audio in an attempt to perform this research, it is not possible to invest the manpower to analyze enough audio to produce statistically meaningful results. This is where the possibility of AI and model training comes in. We decided to research a proof of concept that a model could be effectively trained to classify Spectrogram images and classify what animal produced it. Different models and techniques will have different levels, so being able to tune/use models for different use cases improves efficiency in pursuing/supporting these efforts.

Previous classification models for acoustic data have traditionally converted audio waveforms to images of spectrograms, which has proven to be successful for longer end-to-end audio scene classification. [1] Spectrograms are typically chosen for these longer audio data because they can condense long-term and high-frequency audio data into a much smaller matrix of long term spectral averages (LTSA). However, for audio lengths on the order of 100,000 samples or less (5 seconds at 20kHz audio) a highly detailed spectrogram will contain roughly the same number of data points as the waveform itself.

Additionally, some bioacoustics research has begun to use spectral probability density graphs to identify the presence or absence of animals in a soundscape. By using microphones with onboard processing capabilities, high-frequency audio data can be effectively compressed even further into

Preprint. Under review.

a matrix of size 9xN, where N is the Nyquist frequency (sampling rate divided by 2) of the audio recording [5]. Current research in AI audio classification has yet to incorporate this representation in the machine learning process, which may prove useful for data analysis on large-scale bioacoustic sampling efforts like SanctSound (NOAA).

Our goal in this experiment is to qualify the success of traditional machine learning models in learning the classification boundaries between different animal sounds using various audio representations.

2 Experimental Design

Classification Goal In this experiment, the scope of the research is based on identification of the classification goal. For this, we chose to classify distinct animals based on species. We were able find an appropriate public dataset via GitHub that consisted of 875 distinct audio files of ten different animal species' vocalizations.

Data Preprocessing For the development of this model, we first had to do audio preprocessing to ensure that all of the audio samples had the same peak audio level and sampling rate. By normalizing the peak audio level (i.e. volume), we can avoid the model underfitting the classification pattern based solely on volume. Normalizing the sampling rate of the audio allows all of the spectrograms to have the same upper frequency bound, allowing all of the images being processed to have the same dimensions. Completing these two preprocessing steps ensures that all of the audio files have roughly the same fundamental acoustic properties, and eliminates many possible extraneous variables.

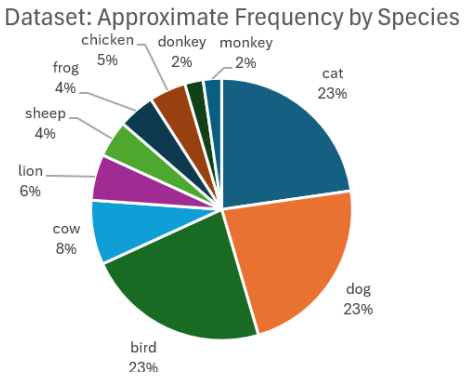

Figure 1: Breakdown of the frequencies of each species in the dataset ($\Sigma = 875$.)

Feature Engineering Standardized audio files were converted into spectrograms using a short-time fourier transformation (STFT) with the Python library, Parselmouth [3]. These spectrograms were processed as matrices using NumPy [2], wherein fixed percentiles (1,5,10,25,50,75,90,95,99) were extracted for each frequency bin.

2.1 Model Architectures

We have chosen to use convolution neural networks (CNNs) for large-scale data processing in this experiment because they are optimal for learning contour-based features of vectors and images without the need for additional feature engineering beyond the convolutions, similarly to the function of human vision-recognition. [4]

Waveforms For identifying waveforms, our data had a fixed sampling rate of 22.05 kHz and one audio channel, and therefore waveform data for a T_w-length audio file was represented as a vector

$$V \in [0, 255]^{1 \times 22050 T_w}$$

To classify data of this shape, we implemented a 1D Convolutional Neural Network with two 1DConv steps and two fully connected layers.

Spectrograms For identifying spectrograms, because our data was sampled at 22.05 kHz, the maximum guaranteed-accurate frequency extracted is the Nyquist frequency $\frac{f_s}{2} = 11.025$ kHz. Because this is still too unwieldy to process, each frequency bin was grouped together in even 10-Hz increments. Additionally, each audio file's time dimension was combined into larger time bins (windows) of length 25 ms. Every input photo was therefore interpreted as a grayscale-matrix

$$S \in [0..255]^{1102 \times \lfloor \frac{T_w}{0.025} \rfloor}$$

To classify these images, we used a traditional 2D convolutional neural network with two convolutions and two fully connected linear layers.

Spectra Spectral percentiles are represented with the same number of columns as there are rows in the spectrogram representation, since each frequency bin is represented as a separate vector. In this study, we use nine percentiles. The values in the matrix are represented in dB (from microphone voltage). Therefore, every spectral-percentile input used for this clsasifier was of size

$$P \in [-70, 70]^{9 \times 1102}$$

To classify these matrices, we used an identical model structure to the spectrogram classification: 2D convolutional neural network with two convolutions and two fully connected linear layers.

3 Results and Discussion

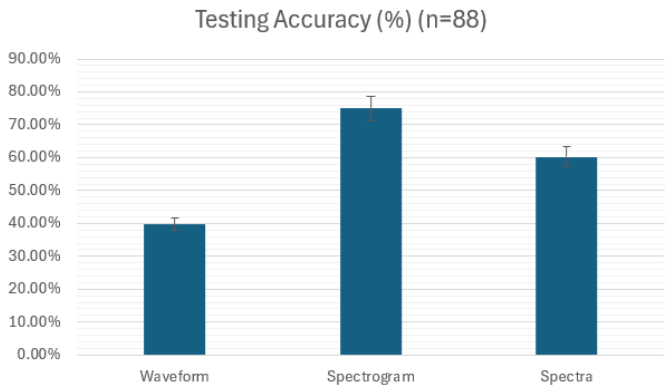

Figure 2: Testing accuracy of the three different types of audio feature manipulations.

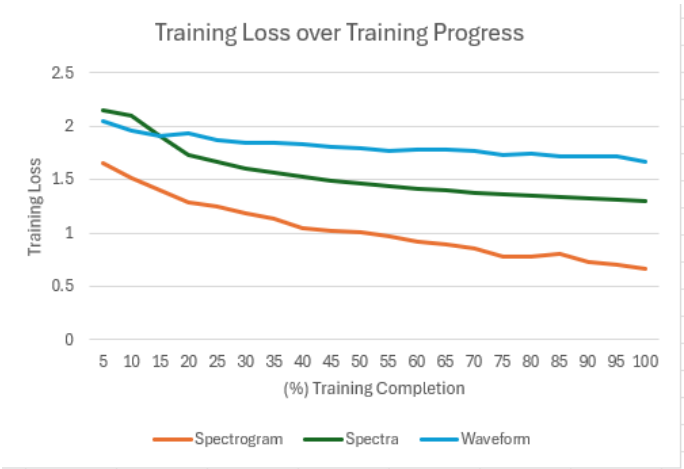

Figure 3: Training loss over the training progress (%).

SPECTRA	Prediction										
Ground truth	dog	cat	bird	cow	monkey	lion	frog	sheep	chicken	donkey	
dog	10	6	1	2	1	0	0	0	0	0	20
cat	3	13	0	2	0	0	0	0	1	1	20
bird	1	1	16	0	1	0	1	0	0	0	20
cow	3	0	0	4	0	0	0	0	0	0	7
monkey	1	0	0	0	1	0	0	0	0	0	2
lion	1	1	0	0	0	3	0	0	0	0	5
frog	1	0	0	1	0	0	2	0	0	0	4
sheep	1	1	0	0	0	0	0	2	0	0	4
chicken	2	0	0	0	1	0	0	0	1	0	4
donkey	0	0	0	1	0	0	0	0	0	1	2
	23	22	17	10	4	3	3	2	2	2	

SPECTROGRAM	Prediction										
Ground Truth	dog	cat	bird	cow	monkey	lion	frog	sheep	chicken	donkey	Grand Total
dog	14	1	0	0	1	4	0	0	0	0	20
cat	2	16	2	0	0	0	0	0	0	0	20
bird	0	0	20	0	0	0	0	0	0	0	20
cow	0	0	0	5	0	0	0	0	1	1	7
monkey	0	0	1	0	0	1	0	0	0	0	2
lion	1	1	0	0	0	3	0	0	0	0	5
frog	0	0	0	0	1	0	1	0	1	1	4
sheep	0	1	0	0	0	0	1	2	0	0	4
chicken	0	0	0	0	0	1	0	0	3	0	4
donkey	0	0	0	0	0	0	0	0	0	2	2
Grand Total	17	19	23	5	2	9	2	2	5	4	88

WAVEFORM	Column Labels										
Row Labels	dog	cat	bird	cow	monkey	lion	frog	sheep	chicken	donkey	Grand Total
dog	3	12	5	0	0	0	0	0	0	0	20
cat	0	13	1	1	1	2	0	0	1	1	20
bird	2	4	9	1	2	0	0	0	0	2	20
cow	0	5	0	2	0	0	0	0	0	0	7
monkey	0	1	0	0	1	0	0	0	0	0	2
lion	0	2	0	0	0	3	0	0	0	0	5
frog	0	2	1	0	0	0	1	0	0	0	4
sheep	0	2	0	0	1	0	0	1	0	0	4
chicken	0	2	0	0	0	0	0	0	2	0	4
donkey	0	1	0	0	0	0	0	0	0	1	2
Grand Total	5	44	16	4	5	5	1	1	3	4	88

Figure 4: Prediction versus Ground Truth confusion matrices for the three different manipulations.

Results illustrate a significant difference between each of the three extracted feature types in terms of test set performance. Spectrograms achieved 75% accuracy over the test set. Additionally, based on the training loss, it seems as though the spectrogram and the spectra achieved the same amount of training-loss reduction overall, but the spectrogram architecture may have been more appropriate for solving the task at hand. However, given the shape of the data, it can also be inferred that there is a significant amount of promise in audio recognition for this significantly compressed spectrogram.

The percentile spectrogram is essentially an extraction of the most important and visually noticeable parts of a normal spectrogram, which is why there might be almost the same amount of meaningful information contained within it. When experts look at spectrograms, they are looking at the contours and significant changes for presence/absence of certain species: thus, an image recognition system for both, in theory, should learn the patterns. In our case, the percentile matrices showed tremendous promise, and with more architecture engineering, might be a very efficient and cost-effective means of storing, transporting, and analyzing long-term and high-frequency audio data.

Of note is that the waveform processing is significantly worse than the other two extracted/engineered feature representations. We believe that this is a result of the high scale of the data and the sheer number of parameters. Beginning with more than 50,000 floating-point numbers representing a pressure-versus-time graph is a large amount of data at scale, and cannot be meaningfully understood without a significant amount of time, energy, and implementation of deep neural networks. In this study, the waveform classification model was trained on a dataset of 875 items using a NVIDIA V100 GPU; nevertheless, the process took nearly sixteen hours to train and test. Processing raw waveforms is tremendously difficult because the data is extremely stochastic and lengthy, and developing a model for it takes an incredible amount of time and energy, and is thus unhelpful.

4 Conclusion

The results show that of the three methods, the general most-effective application of a convolutional neural network on audio is for a spectrogram.

5 References

[1] Eghbal-Zadeh, H., Lehner, B., Dorfer, M., & Widmer, G. (2016). CP-JKU Submissions for DCASE-2016: A Hybrid Approach Using Binaural I-Vectors and Deep Convolutional Neural Networks. Detection and Classification of Acoustic Scenes and Events 2016. https://www.cp.jku.at/research/papers/Eghbal-Zadeh_etal_DCASE_2016.pdf

[2] Harris, C. R., Millman, K. J., Van Der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Van Kerkwijk, M. H., Brett, M., Haldane, A., Del Río, J. F., Wiebe, M., Peterson, P., ... Oliphant, T. E. (2020). Array programming with NumPy. Nature, 585(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

[3] Jadoul, Y., Thompson, B., & De Boer, B. (2018). Introducing Parselmouth: A Python interface to Praat. Journal of Phonetics, 71, 1–15. https://doi.org/10.1016/j.wocn.2018.07.001

[4] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444. https://doi.org/10.1038/nature14539

[5] Martin, S. B., Gaudet, B. J., Klinck, H., Dugan, P. J., Miksis-Olds, J. L., Mellinger, D. K., Mann, D. A., Boebel, O., Wilson, C. C., Ponirakis, D. W., & Moors-Murphy, H. (2021). Hybrid millidecade spectra: A practical format for exchange of long-term ambient sound data. JASA Express Letters, 1(1), 011203. https://doi.org/10.1121/10.0003324

[6] Nita, Y. (2018). Animal Sound Dataset (Version 574b5e2) [Dataset]. https://github.com/YashNita/Animal-Sound-Dataset