Audio Embedding Approach - VGGish Model

# Introduction & Context

This work was completed as part of the 2024 University of Washington Master of Science in Data Science Capstone Project. Our team was mutually matched with Dr. Shima Abadi’s team at the University of Washington where we worked closely with Dr. Christopher Basset, Dr. Emma Cotter, and Mitchell Scott.

Our goal was to devise a method to identify WEC sounds in hydrophone recordings taken at the US Navy Wave Energy Test Site (WETS) with the goal of generalizing detection to additional sites. This task is challenging due to the complexity of the acoustic environment, variable nature of WEC sounds, and the inability to know WEC sounds a priori.

As part of the Capstone preparation work completed in Fall 2023, our team identified 4 possible methods of WEC sound detection and chose to execute 2 POCs. One POC explored an audio embedding and classification model combination. We believed this approach was promising because of its unsupervised nature, avoiding the need for lengthy human data annotation and issue of detecting an unknown sound. Initially we proposed trying two audio embedding models, VGGish and Yamnet, but chose to pursue only VGGish given we would also be investigating a separate image detection approach (YOLOv8 One Minus Approach).

There were two highlighted concerns with VGGish. The first was the team’s lack of familiarity with the model, making it difficult to accurately estimate workload and foresee challenges. The second was the model’s training dataset and its transferability to our use case. While VGGish was built to specifically identify and embed spectrogram features, it was trained on YouTube data, which was unlikely to sound like our underwater hydrophone environment (perhaps with the exception of whale calls).

We split the 2024 Capstone into 3 phases, a fall POC, development phase one in mid-January/early February, and development phase two in late February/early March. The POC served to validate our ability to install necessary libraries and run a test set of data through the model. The first development phase allowed the team to customize the model, develop audio resampling scripts, run a preliminary pass of all data using model default parameters, and attempt a first clustering method. The second development phase allowed us to tweak the model, testing multiple example lengths and clustering methodologies, enabling us to select the process which yielded the best results in our selected metrics (Silhouette Score and density of known sounds).

# Data

For this portion of the project, we chose to focus on the hydrophone data gathered between December 1st, 2018 and February 28th, 2019 containing noises from the Fred Olsen. There were 1,325 recordings saved as .wav files, each lasting approximately 30 minutes. From annotations intended for YOLOv8, we knew the data contained notable humpback whale, airplane, fish, boat, helicopter, flow, and mooring noises. We also found each audio file began with ~1 second set of calibration tones, and spectrograms would suggest some recordings captured periodic WEC noises.

The .wav files were originally sampled as 48kHz mono, 32-bit floats. To align with VGGish training conditions and produce a greater likelihood of model effectiveness, we developed a script which would programmatically resample the audio to signed 16-bit PCM at 16kHz mono. The resampling script, 02\_audio\_resampling.py, leverages the python librosa and soundfile libraries. We chose to use the default librosa.resample type, soxr\_hq, which provides a high-quality FFT-based bandlimited interpolation. There were other resampling types we could have chosen, for more information on these resampling techniques, researchers should review the librosa documentation [1].

# Related Work

The embedding and clustering approach was inspired by a talk given by OrcaAL [4], however, our team could not locate published works detailing their theory and methods. We did find [3] useful to validate and further inspire our approach. In the paper, Di et al. noted, “On account of the scale and diversity of the YouTube dataset [used to train VGGish], the resulting acoustic feature are both very general and of high resolution, placing each audio sample in a high-dimensional feature space that is unlikely to show ecosystem-specific bias” [3], reducing our initial concern that VGGish wouldn’t generalize well to hydrophone recordings.

Di et al. attempted to programmatically recognize known sounds from bee colonies to better understand colony circumstances. The authors used VGGish to distill recordings into audio embeddings and compared the effectiveness of t-SNE and UMAP in reducing dimensionality from 128 to 2. Because the authors were using labeled data, they compared supervised data classification methods including decision trees, K-nearest neighbors (KNN), support vector machines and random forests. Ultimately Di et al. found UMAP performed better than t-SNE in separating known bee colony sounds and KNN produced the highest classification accuracy of ~90% across 3 datasets. Given [3] success in identifying and classifying sounds, we propose a similar approach for further research at the end of this report.

Our approach to example duration testing was inspired by the methodologies found in [2]. In [2] the authors attempted to identify/classify sounds and predict their start and end times in audio sessions using convolutional neural networks (CNN). Similar to our data, their recordings possessed sounds lasting variable lengths of time ranging from very short durations to tens of seconds. To explore the model’s classification capabilities and minimum amount of signal required to reliably identify a sound, the researchers explored the effect of different window lengths (2, 5 and 20 seconds) on class outputs and the granularity of start/stop times. As might be expected, 2 and 5 second windows provided the researchers’ models with less context to predict sounds but generated better temporal event resolution. In contract, 20 second windows allowed for more information in class determination but generated coarser estimates of sound start/stop times and class labels.

# Methods & Results

We began by installing all necessary libraries and files as indicated in the VGGish README. As noted in the Data Section above, it was necessary to build a data resampling script to structure the .wav files in the format expected by the model. This script was run on all raw .wav files (‘D:\1Dec2018\_28Feb2019\Hydrophone’) which generated resampled files saved to the external hard drive under ‘D:\1Dec2018\_28Feb2019\Hydrophone\_Resampled’. When attempting to vary the length of audio captured per embedding, the team realized VGGish had a fixed input parameter of 96 features per example. As installed, this would typically be satisfied by 96 log Mel spectrograms each representing 0.01 seconds of audio. We were able to alter the length represented by each feature (log Mel spectrogram) to create longer or shorter embeddings. The formulas required to change embedding lengths can be found in the VGGish README.

Based on conversations with our project sponsor, it was clear that sounds in our underwater environment lasted from a split second (single shrimp snaps) to over a minute (passing boats). Examining spectra used for the YOLOv8 model, we found that many sounds tended to have individual elements lasting one or two seconds which may be repeated over time (e.g., humpback vocalizations or mooring sounds). We borrowed the 2 second and 5 second examples lengths from [2] to test the impact of different embedded audio durations on our clusters.

We used k-means clustering due to our unlabeled data. We expected to find at least 10 clusters, representing the known environmental sounds (humpback whale, airplane, fish, boat, helicopter, flow noise, mooring noise, calibration tones, WEC, and silence), but the exact number audio classes was unknown. We tested two methods of dimensionality reduction, t-SNE and UMAP, per [3] to make results easily visualizable and reduce k-means processing time. By running DBSCAN during dimensionality reduction, we returned a calculated number of clusters present. There are many measurements used to evaluate the “goodness” of clustering including Silhouette Score, inertia, Davies-Bouldin, and Calinski-Harabasz metrics. We chose to use Silhouette Score given its proxy for measuring cohesive and distinct groupings of points.

We had the advantage of possessing annotated spectrograms corresponding to minute-long sections of some .wav files input to VGGish. By tracing annotated sounds back to their time of occurrence, we were able to label some of VGGish’s embeddings. If k-means was working well, we expected like sounds to reside in the same clusters (e.g., all airplane sounds together, all boat sounds together). We developed a heatmap of known sound cluster density which was also used to assess goodness of k-means fit.

We tested 3 embedding lengths (0.96, 2 and 5 seconds) and two methods of dimensionality reduction (t-SNE and UMAP) to study their impact on k-means cluster formation. We found that UMAP consistently produced fewer clusters than expected given audio annotations, so we chose not to pursue its results. Of the three embedding lengths, 5 seconds produced the highest t-SNE Silhouette Score.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Embedding Length (sec) | Number of clusters | | Silhouette Score | |
| t-SNE | UMAP | t-SNE | UMAP |
| 0.96 | 12 | 4 | 0.3515 | N/A |
| 2.0 | 18 | 3 | 0.3447 | N/A |
| 5.0 | 32 | 2 | 0.3527 | N/A |

Table 1. Numbers of clusters and Silhouette Scores by embedding length and dimensionality reduction method.

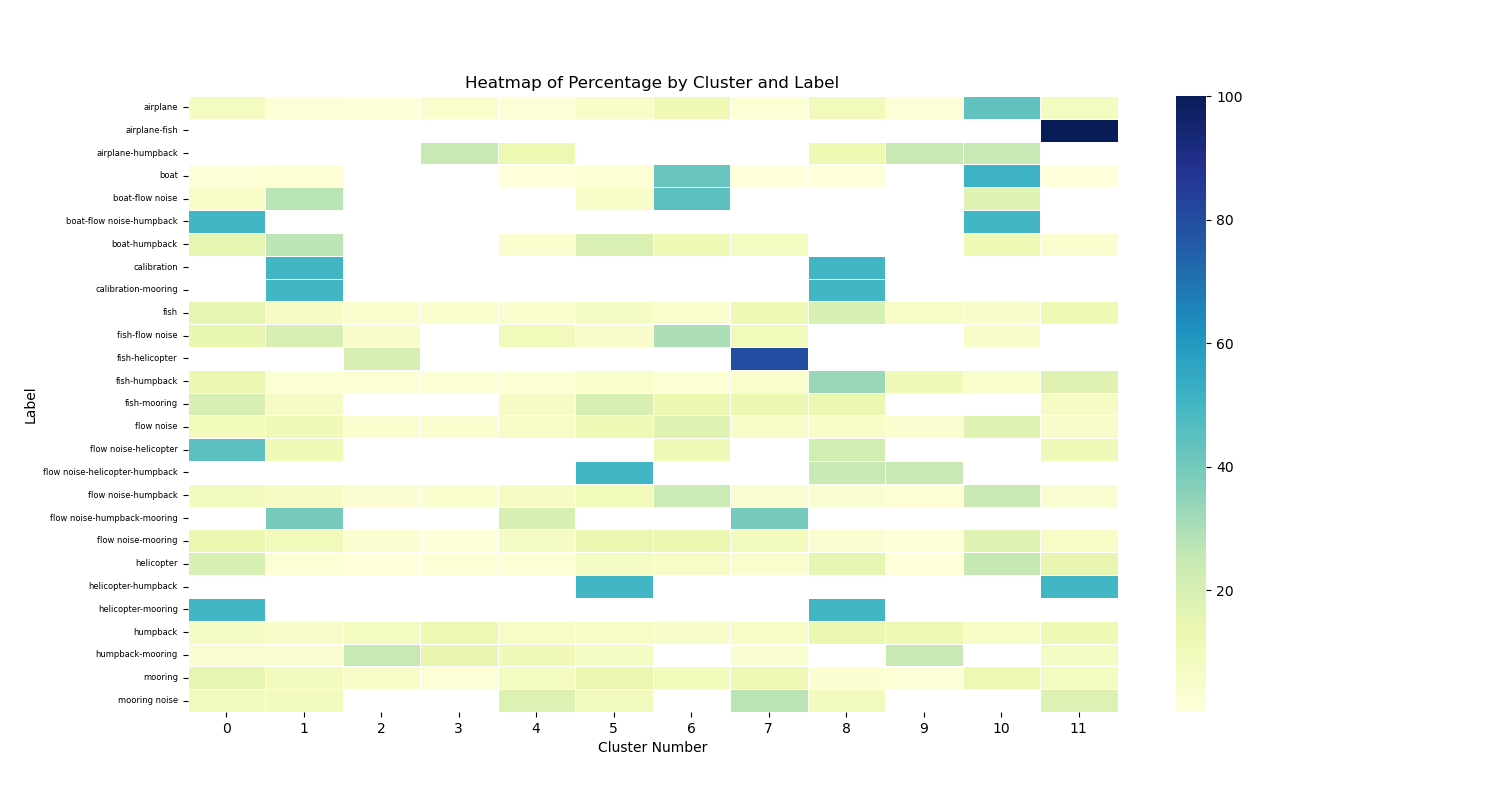
****

Figure 1. 0.96 second embedding heatmap of known sounds in clusters

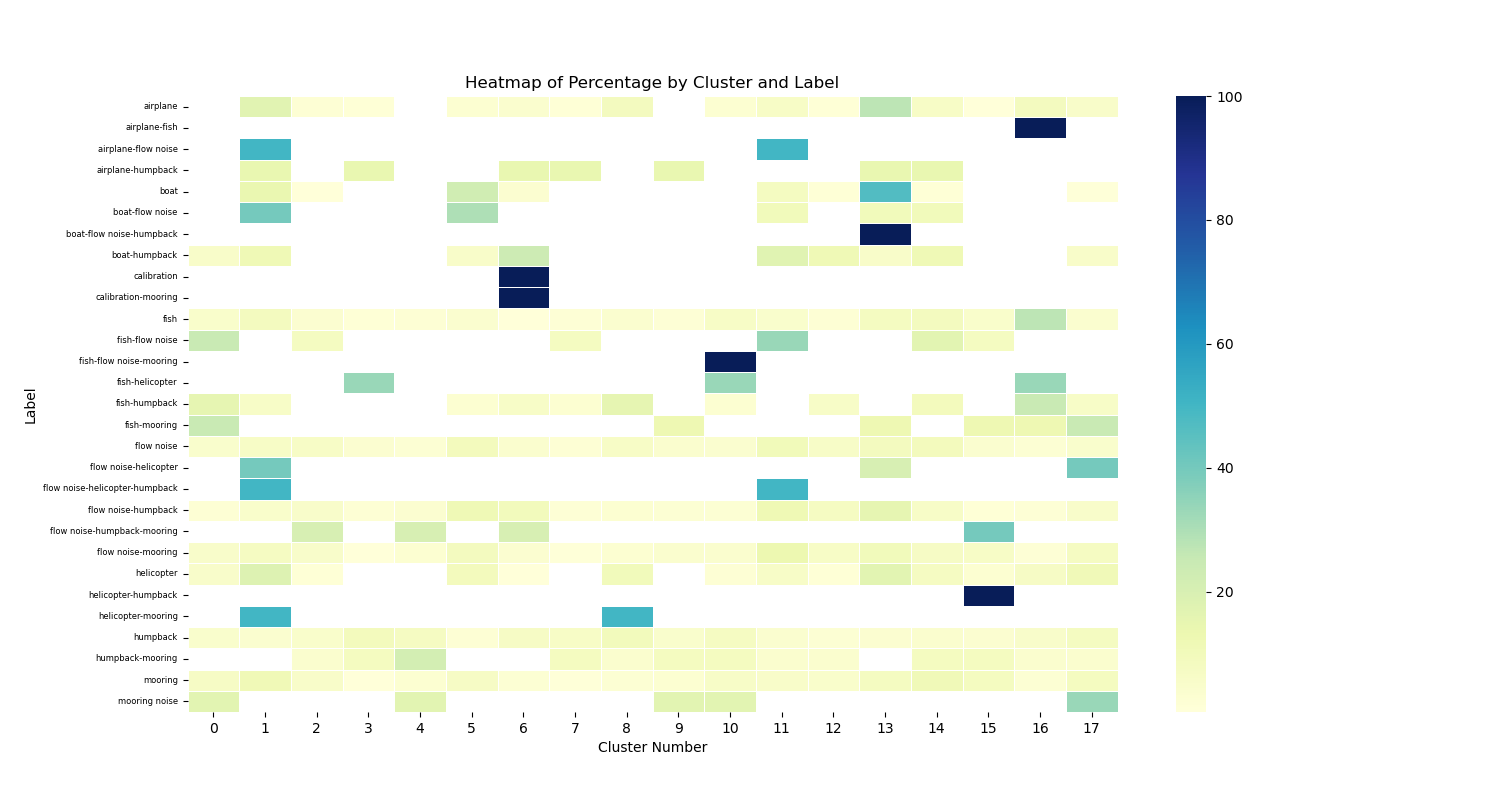


Figure 2. 2 second embedding heatmap of known sounds in clusters

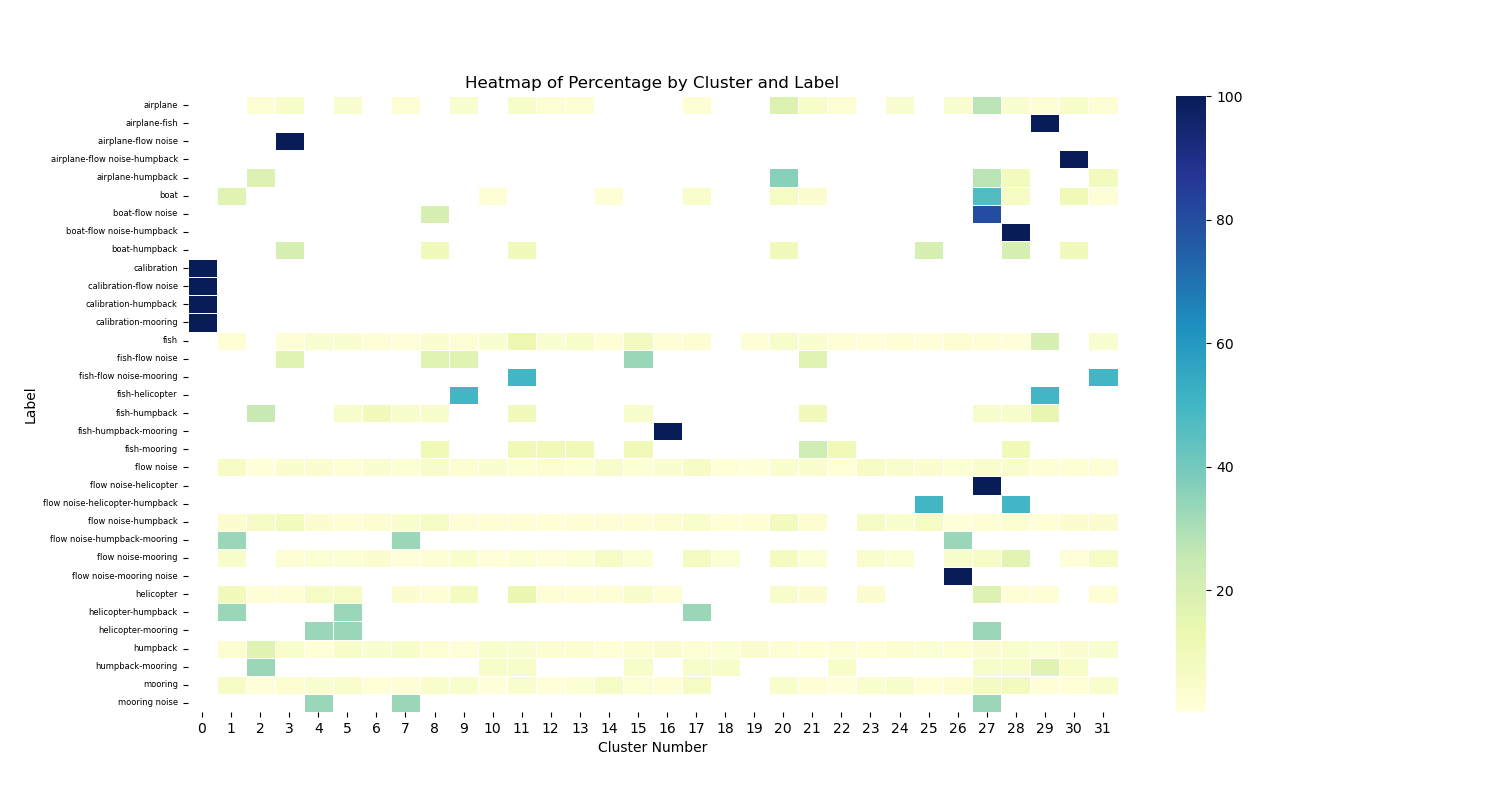


Figure 3. 5 second embedding heatmap of known sounds in clusters

Visually comparing the three heatmaps of known sound densities above, we see that 5-second embeddings produced the greatest number of clusters containing ~100% of a known sound. We hypothesize that airplanes, boats, and calibration sounds produce cohesive groupings because of their distinctness and ability to mask all other sounds in the audio recording. We are unsurprised to see humpback embeddings spread across many clusters due to the sheer variety of whale sounds. It would be interesting to test how clustering results changed if humpback calls were annotated to name each kind of sound individually.

Ultimately, we found that 5-second embeddings using t-SNE yielded both the highest Silhouette Score and density of known sounds. Segmenting audio into 5-second samples likely captured sounds in their entirety, encoding them into single embeddings, rather than artificially splitting up the sound. This approach also reduced the presence of noisy data caused by various compositions of underwater "silence".

Examining the clusters output from t-SNE and k-means using 0.96, 2, and 5 second embeddings, we can see that none produce distinctly grouped clusters. Silhouette Scores range from -1 to 1 with 1 indicating well-separated, distinct clusters and -1 suggesting incorrectly assigned clusters. Our scores of ~0.35 shows clusters trending in the right direction, but still with significant overlap, as is reflected in the images below. Interestingly, calibration sounds tend to produce the most distinct clusters, potentially due to their volume and difference from other sounds.

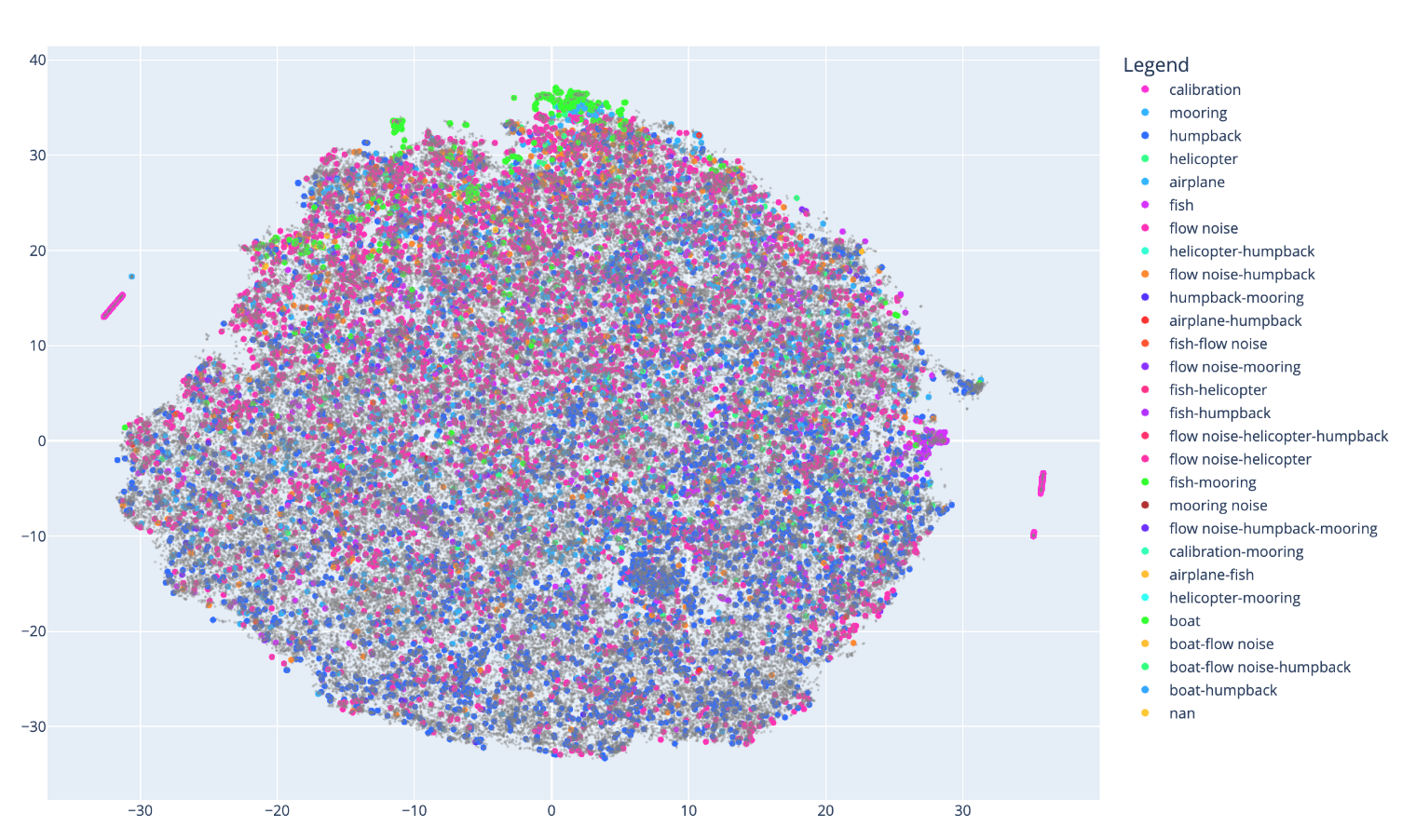


Figure 4. 0.96 second embedding clusters

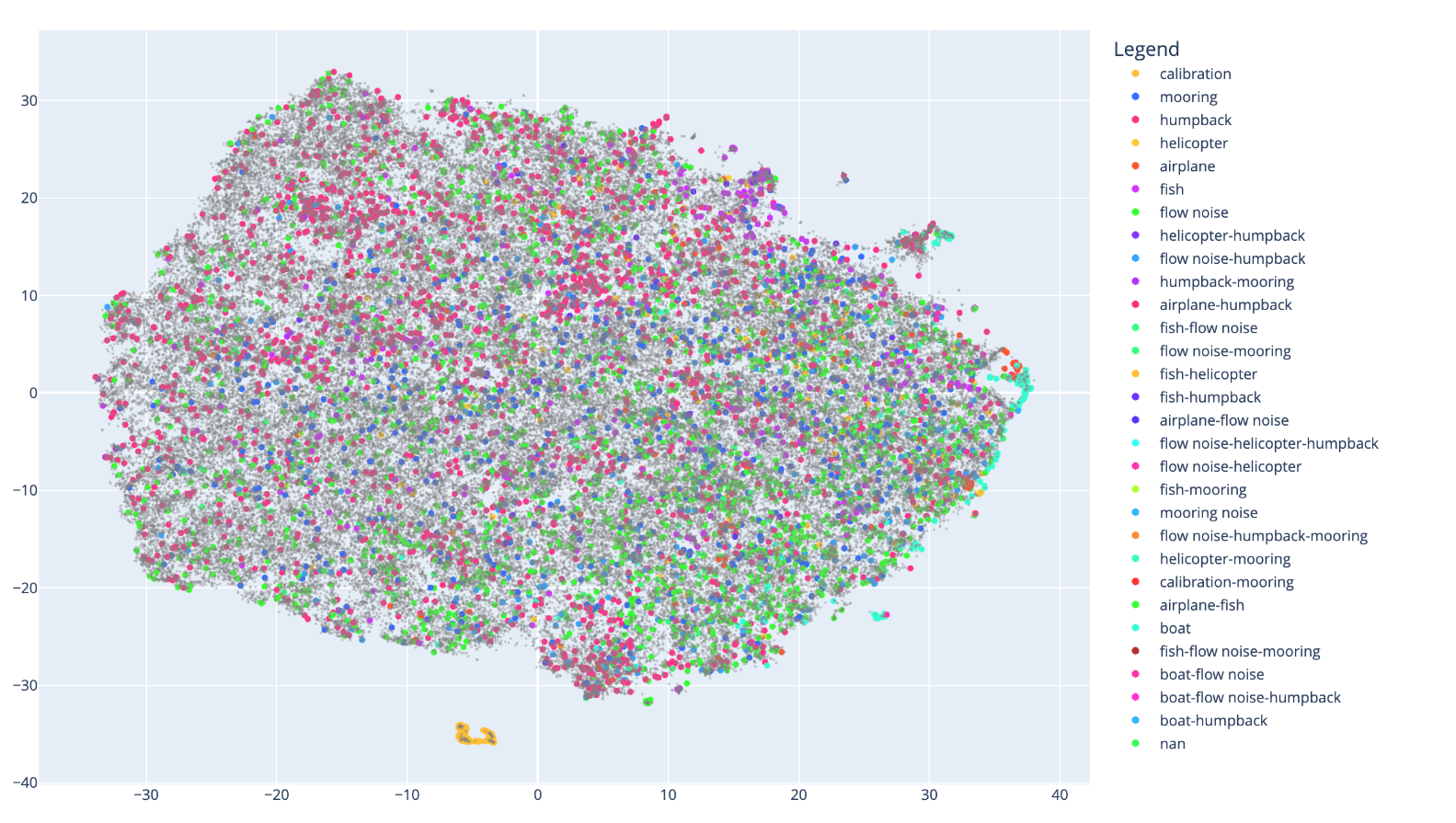


Figure 5. 2 second embedding clusters

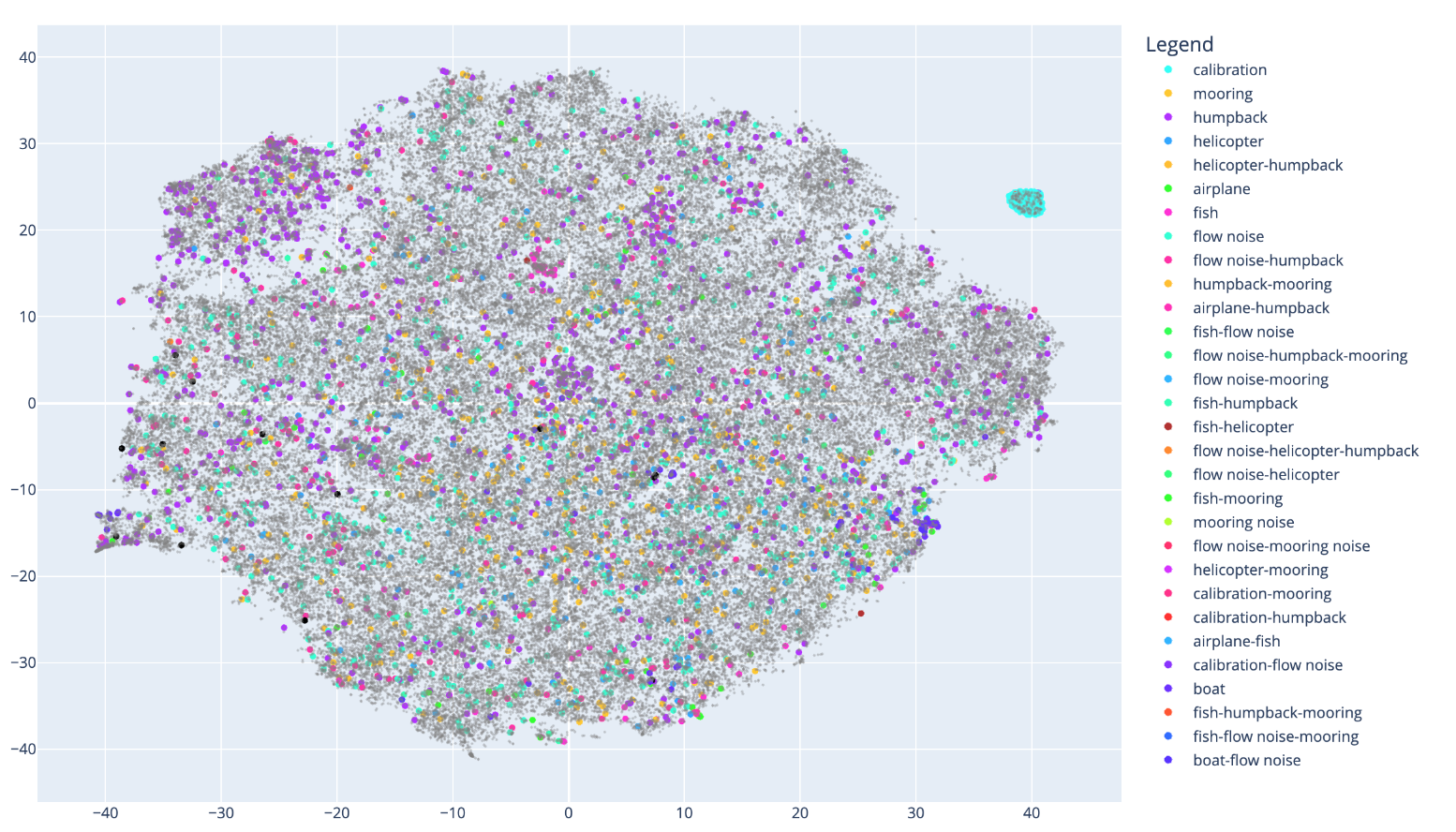


Figure 6. 5 second embedding clusters

If clusters had been well separated, we would have been able to confidently identify those with no known sound labels, potentially signaling the embedded audio segments contained WEC or other Mapping sound embeddings to times of occurrence, a marine acoustics specialist could have listened to the original audio files and provided correct embedding labels. Well-defined clusters would have also allowed us to more confidently assume that embeddings in the same cluster contained the same sounds - dramatically reducing the time needed for manual identification and annotation.

# Conclusions

While this is an interesting and novel approach, the lack of clear clusters containing annotated sounds indicates it may not be well-suited to acoustically complex environments without further development.

It is possible that other audio resampling methods, or no resampling, could generate richer .wav files and potentially cleaner embeddings and clusters. In alignment with [3], one could turn this into a supervised learning problem, using only .wav files with annotated sounds. By classifying embeddings as a known sound, silence, or unknown sound, one could potentially generate more accurate results for sorting and identifying unseen sounds. Finally, it could be interesting to test other methods of dimensionality reduction or audio embedding which may better retain or represent the relationships between different sounds.

# References

1. Anon. Librosa.resample. Retrieved March 10, 2024b from <https://librosa.org/doc/main/generated/librosa.resample.html>. DOI 10.5281/zenodo.8252662
2. Malek Ibrahim, Jason Sagers, Megan Ballard; A convolutional neural network applied to Arctic acoustic recordings to identify soundscape components. Proc. Mtgs. Acoust. 11 December 2020; 42 (1): 070005. <https://doi.org/10.1121/2.0001393>
3. Nayan Di​, Muhammad Zahid Sharif, Zongwen Hu, Renjie Xue, and Baizhong Yu. 2023. Applicability of vggish embedding in bee colony monitoring: Comparison with MFCC in colony sound classification. (January 2023). Retrieved March 10, 2024 from https://peerj.com/articles/14696/
4. Valentina Staneva.Retrieved March 10, 2024b from https://static.us.edusercontent.com/files/6HSFqWFYrwDqGaZJqxn8HNoN