Few-Shot Learning for Audio Classification with HuBERT

Ilan Bacry
ENS Paris-Saclay
Paris, France
ilan.bacry20@gmail.com

Théo Basseras
ENS Paris-Saclay
Paris, France
ilan.bacry20@gmail.com

Abstract

Foundation models for speech provide powerful representations that can be adapted to a wide range of downstream tasks, but the computational and data costs associated with supervised fine-tuning can be substantial. In contrast, few-shot learning offers a lightweight alternative that requires only a small number of labeled examples and minimal additional training. In this work, we design a set of experiments comparing few-shot learning and supervised fine-tuning as adaptation strategies for HuBERT [4], a self-supervised foundation model for speech. We use fine-tuning as a reference point to assess how closely few-shot learning can approach its performance across a diverse collection of audio classification datasets, varying in scale, complexity and acoustic diversity. By framing fine-tuning as a resource-intensive baseline, our experimental setup aims to analyze the trade-offs between adaptation cost and predictive accuracy when deploying foundation models in low-resource and data-scarce scenarios.

1 Introduction

Self-supervised learning has become a central paradigm for learning general and transferable representations from large amounts of unlabeled data. In natural language processing, models such as BERT [2] have demonstrated that pretraining on large corpora yields representations that can be reused across many downstream tasks, laying the groundwork for what are now commonly referred to as foundation models. This paradigm has been successfully extended to speech with models such as HuBERT [4], which learn high-level speech representations without relying on transcriptions. By capturing both acoustic and linguistic structure, HuBERT can be viewed as a speech foundation model: a single pretrained model that supports adaptation to a variety of audio tasks with limited task-specific supervision.

Adapting foundation models to downstream tasks is most commonly achieved through supervised fine-tuning. While effective, fine-tuning often requires substantial computational resources and labeled data, which can be limiting in practice, especially when dealing with diverse datasets or low-resource settings. Few-shot learning therefore emerges as a lightweight alternative, aiming to leverage foundation model representations using only a small number of labeled examples.

This report is organized as follows. First, a brief overview of BERT [2] is provided, HuBERT [4] is then presented as its speech-domain counterpart. Next, the few-shot learning framework used in this study is described, detailing how few-shot classification is constructed and applied to pretrained speech representations. Finally, the experimental comparison between few-shot learning and supervised fine-tuning is presented across a range of audio classification tasks. Our code is publicly available at https://github.com/luckyman94/Few-shot-learning-with-HuBERT

2 From BERT to HuBERT

2.1 BERT

BERT [2] is based on a stack of Transformer encoders trained with a masked prediction objective, where randomly masked tokens are predicted from their surrounding context. By leveraging both left and right contexts, this approach enables BERT to learn rich bidirectional representations that capture contextual and semantic information.

2.2 HuBERT

HuBERT [4] extends the masked prediction paradigm introduced by BERT to the speech domain. Instead of operating on discrete text tokens, HuBERT learns from raw audio by predicting masked latent representations derived from speech signals. This allows the model to acquire high-level and transferable speech representations without relying on transcriptions.

2.2.1 Architecture

As illustrated in Figure 1, HuBERT is composed of three main components :

- **Convolutional feature encoder.** Raw audio is first processed by a convolutional neural network that extracts latent speech representations.
- **Transformer encoder.** Inspired by BERT, a stack of Transformer layers models long-range temporal dependencies and contextual information using a masked prediction objective.
- **Acoustic unit discovery system.** Discrete acoustic units are obtained through unsupervised clustering and used as prediction targets during training, enabling fully self-supervised learning without transcriptions.

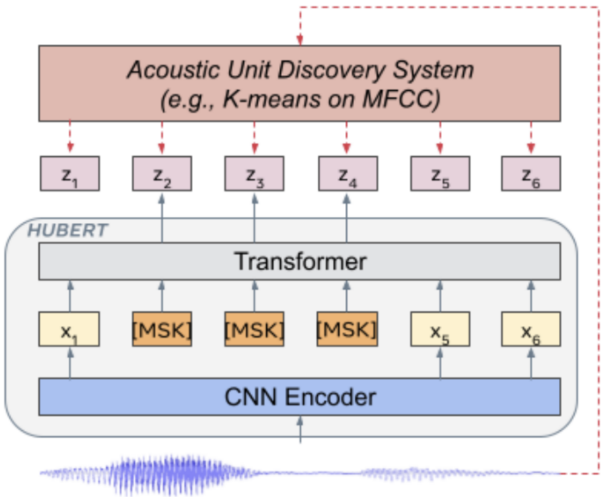

Figure 1: Overview of the HuBERT architecture.

3 Few-Shot Learning

3.1 Formulation

Few-shot learning aims to address classification problems in which only a limited number of labeled examples per class are available. In this setting, models must rapidly adapt to new tasks given minimal supervision.

In this work, we consider the standard k-shot, n-way episodic learning setting.

Definition We define an *episode* in the context of k-shot, n-way learning, where n denotes the number of distinct classes involved in the task. Each episode is composed of two disjoint subsets: a *support set* and a *query set*.

- **Support set.** For each of the n classes, k labeled examples are randomly sampled, resulting in a total of kn support samples.

- **Query set.** For each class, mk additional examples are sampled, ensuring that these samples are distinct from those used in the support set.

The objective of few shot learning is to classify the query samples using only the information provided by the support set.

An overview of the episodic few-shot learning setting is illustrated in Figure 2.

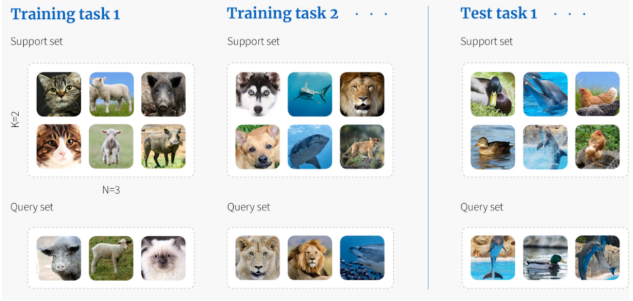

Figure 2: Illustration of the episodic few-shot learning setting. Each task is composed of a support set and a query set.

In the following, we describe the prototypical few shot learning [7] strategy adopted in this work to perform few-shot learning with HuBERT representations.

3.2 Prototypical Few Shot Learning

We again consider the k-shot, n-way setting. Each episode is processed in two main phases :

- **Support phase.** We construct a support set containing n classes with k samples per class:

$$(s_1^1, \ldots, s_k^1, s_1^2, \ldots, s_k^n).$$

For each class j, the k samples are projected into a latent space using the projected HuBERT-based embedding function:

$$l_i^j = \text{HuBERT}_{\text{projected}}(s_i^j), \quad i = 1, \ldots, k.$$

The prototype for class j is then computed as the mean of its embeddings:

$$r_j = \frac{1}{k} \sum_{i=1}^{k} l_i^j.$$

Each prototype r_j represents its corresponding class in the latent space. Classification is then performed by assigning each query sample to the class whose prototype is closest in the embedding space.

- **Query phase.** We construct a query set containing the same n classes, with km samples per class:

$$(s_1^1, \ldots, s_{km}^1, s_1^2, \ldots, s_{km}^n).$$

Each query sample s_i^j is projected into the latent space:

$$l_i^j = \text{HuBERT}_{\text{projected}}(s_i^j).$$

For each query embedding l_i^j, we compute its squared Euclidean distance to all class prototypes:

$$d_i^j = \left[\|r_1 - l_i^j\|^2, \ldots, \|r_n - l_i^j\|^2\right].$$

A softmax over the negative distances yields a probability distribution over classes:

$$p_i^j(c) = \frac{\exp\left(-\|r_c - l_i^j\|^2\right)}{\sum_{h=1}^{n} \exp\left(-\|r_h - l_i^j\|^2\right)}, \quad c = 1, \ldots, n.$$

The predicted class is given by:

$$\hat{y}_i^j = \arg \max_{c \in \{1,\dots,n\}} p_i^j(c).$$

Figure 3 provides a concise overview of the proposed few-shot classification pipeline.

Figure 3: Overview of the prototypical few-shot classification pipeline with a frozen HuBERT encoder.

Note : In our experimental setting, the datasets under study contain a relatively small number of classes. As a result, we always consider the *full* classification problem and fix n-way to be equal to the total number of classes in the dataset. Consequently, n is not varied across episodes, and few-shot learning is evaluated by varying only the number of support examples per class (k).

Practical setting While the episodic few-shot framework described above is presented in its general form, our experimental setting follows a simplified but realistic configuration. Since the datasets under study contain a relatively small number of classes, we always consider the full classification problem and fix the n-way setting to be equal to the total number of classes in the dataset. As a result, n is not varied across episodes and few-shot learning performance is evaluated by varying only the number of labeled support examples per class (k).

4 Experimental Setup

4.1 Datasets

The experiments were conducted on a diverse collection of audio classification datasets, designed to evaluate few-shot learning under a wide range of conditions. We consider two main categories of datasets.

First, we use real-world audio classification datasets collected from publicly available sources, primarily Kaggle. These datasets correspond to standard and well-established audio classification tasks. Depending on the dataset, we optionally apply balanced sub-sampling in order to control the number of classes and samples per class and to ensure fair and comparable few-shot evaluation.

Second, we consider synthetic audio datasets specifically designed to probe the behavior of few-shot learning under controlled conditions. These datasets are used to isolate specific factors and to analyze model behavior in simplified settings. Details regarding their construction are provided later.

The datasets used in our experiments are presented in detail below.

4.1.1 Real-World Audio Classification Datasets

Speech Commands [9] Speech Commands is a keyword recognition dataset composed of short, isolated spoken words. It represents a simple and well-separated classification task and serves as an entry-level benchmark. This dataset is a available on Kaggle.[1]

CREMA-D [1] CREMA-D is an emotional speech dataset where actors express predefined sentences with different emotions. The task involves finer-grained acoustic distinctions. This dataset is available on Kaggle.[2]

TIMIT [3] TIMIT is a phoneme-level speech dataset that requires discrimination between fine-grained acoustic units. This dataset introduces higher task complexity despite its relatively small size. This dataset is available on Kaggle.[3]

Cats vs. Dogs vs. Birds An audio-based multiclass classification dataset of cat, dog and bird sounds. The Cats vs Dogs vs Birds dataset is a publicly available audio dataset released on Kaggle.[4]

Snoring Detection A binary audio classification dataset composed of snoring and non-snoring sounds. The Snoring dataset is a publicly available dataset released on Kaggle and does not have an associated peer-reviewed publication.[5]

UrbanSound8K [6] A multi-class urban sound classification dataset with ten everyday sound categories. This dataset is available on Kaggle.[6]

4.1.2 Synthetic Datasets

Synthetic Audio Noise A synthetic audio classification dataset composed of multiple classes of sinusoidal signals corrupted by additive Gaussian noise at a fixed signal-to-noise ratio. Each class is associated with a distinct base frequency, and the dataset is balanced with fixed-length audio samples.

Synthetic Audio Harmonics A synthetic audio classification dataset where each class corresponds to a signal composed of a fundamental frequency and a varying number of harmonic components with decreasing amplitudes.

A summary of the main characteristics of all datasets used in our experiments is provided in Table 1.

[1] https://www.kaggle.com/datasets/nikhilkushwaha2529/speech-commands
[2] https://www.kaggle.com/datasets/ejlok1/cremad
[3] https://www.kaggle.com/datasets/nltkdata/timitcorpus
[4] https://www.kaggle.com/datasets/warcoder/cats-vs-dogs-vs-birds-audio-classification
[5] https://www.kaggle.com/datasets/tareqkhanemu/snoring
[6] https://www.kaggle.com/datasets/chrisfilo/urbansound8k

Table 1: Summary of datasets used in the experiments.

Dataset	Classes	Samples
Speech Commands	7	1000
CREMA-D	6	7442
TIMIT	10	1000
Cats vs. Dogs vs. Birds	3	610
Snoring Detection	2	1000
UrbanSound8K	10	8732
Synthetic Audio Noise	8	800
Synthetic Audio Harmonics	8	800

4.2 Protocol and Training Parameters

4.2.1 Few Shot Learning Protocol

Few-shot experiments are evaluated exclusively on the audio classification datasets described above. In this section, we present the few-shot benchmarking protocol adopted in our experiments, along with the associated experimental settings.

- **Task type:** Binary and multi-class audio classification.
- **Query set:** 20
- **Evaluation metric:** Classification accuracy,

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN},$$

- **Few-shot regimes:** 1-shot, 5-shot and 10-shot settings.
- **Distance metric:** Euclidean distance in the embedding space, chosen as it captures both magnitude and direction, whereas cosine similarity focuses only on angular information.

Few-shot benchmarking is performed by repeatedly sampling a fixed number of episodes for each dataset and each k-shot setting. Specifically, experiments are conducted for $k \in \{1, 3, 10\}$. For each episode, the support set contains k labeled examples per class, while the query set is composed of a dataset-dependent number of samples per class drawn from the remaining data. The number of classes involved in each task also depends on the dataset under study.

Classification accuracy is computed on the query set for each episode. Final few-shot performance is reported as the mean accuracy and standard deviation across all sampled episodes, providing a robust estimate of performance under low-resource conditions.

After benchmarking, confusion matrices and t-SNE [8] are computed and plotted to further analyze class-wise performance and error patterns.

4.2.2 Fine-Tuning as a Reference

For three datasets, *Cats vs. Dogs vs. Birds*, *Snoring Detection* and *UrbanSound8K*, supervised fine-tuning of the HuBERT foundation model is performed in order to establish a reference point for comparison with few-shot learning. These datasets are deliberately chosen to cover different levels of task complexity, ranging from a simple and well-separated classification problem, to an intermediate binary task and finally to a more challenging multi-class real-world audio classification task. The fine-tuning procedure follows the protocol described below:

- **Data splits:** Each dataset is split into training, validation and test sets using stratified sampling, with proportions of $70\%/15\%/15\%$, respectively.
- **Batch size:** A batch size of 16 or 8 (depends on the dataset) is used.
- **Classification head:** A lightweight linear classification head is added on top of HuBERT representations, using mean pooling over the temporal dimension.
- **Training epochs:** The number of training epochs varies between 8 and 10 depending on the dataset.
- **Loss function:** Cross-entropy loss $\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log(\hat{y}_{i,c})$.

Moreover, due to the large size of the HuBERT model, fine-tuning is performed using parameter-efficient LoRA [5] adapters rather than updating all model parameters. This approach allows us to obtain a strong fine-tuning reference while keeping the computational cost manageable. For all our experiences, the LoRA configuration was : *Rank* : 4, α : 8, *Dropout* : 0.2

5 Results and Discussion

5.1 Few-shot Learning

Dataset	1-shot	3-shot	10-shot
Speech Commands	55.7	81.1	86.4
CREMA-D	26.6	33.1	36.8
TIMIT	42.0	62.2	72.3
Cats vs. Dogs vs. Birds	72.8	92.1	95.4
Snoring Detection	55.0	62.8	66.8
UrbanSound8K	20.7	22.7	22.1
Synthetic Noise (Low)	99.4	99.7	99.9
Synthetic Noise (Mid)	96.9	98.3	98.4
Synthetic Noise (High)	91.1	95.5	96.2
Synthetic Harmonics (Low)	99.4	99.7	99.9
Synthetic Harmonics (Mid)	72.2	70.6	69.8
Synthetic Harmonics (High)	1.0	1.0	1.0

Table 2: Few-shot accuracy averaged over 100 episodes for each k-shot setting. Synthetic datasets are evaluated at three levels of difficulty (Low, Mid, High).

Additional qualitative analyses, including confusion matrices and t-SNE [8] visualizations for selected datasets, are reported in Appendix A.

For the synthetic datasets, *Low*, *Mid* and *High* denote increasing levels of task difficulty. For the Synthetic Noise dataset, these levels correspond to decreasing signal-to-noise ratios (*Low*: 30 dB, *Mid*: 10 dB, *High*: 0 dB). For

the Synthetic Harmonics dataset, difficulty is controlled by the number of harmonic components present in the signal (*Low*: 2, *Mid*: 5, *High*: 10, capped by the number of available classes).

Table 2 presents few-shot classification accuracy across datasets under different k-shot settings. A clear and consistent trend can be observed across all datasets: performance improves as the number of labeled examples per class increases from 1-shot to 10-shot, highlighting the strong dependence of few-shot learning on the amount of available supervision.

In the 1-shot regime, simpler and more controlled datasets achieve higher accuracy, indicating that HuBERT representations are sufficiently discriminative to support classification from a single labeled example. For instance, datasets such as *Speech Commands* and *Cats vs. Dogs vs. Birds* exhibit strong performance in the 1-shot setting, reflecting well-separated classes and limited intra-class variability.

In contrast, more complex datasets, including *CREMA-D* and *UrbanSound8K*, exhibit lower performance in the 1-shot regime due to higher acoustic variability and more subtle class boundaries.

Moving from 1-shot to 5-shot yields a noticeable performance improvement on most datasets, as additional support examples lead to more reliable prototype estimation. This trend is observed, for instance, on *TIMIT* and *CREMA-D*, where increased supervision helps reduce intra-class variability.

However, the improvement is less pronounced for more complex datasets such as *UrbanSound8K*, where high acoustic diversity limits the benefit of a small increase in support samples.

Finally, the 10-shot setting further improves performance by stabilizing class prototypes, although gains tend to saturate on simpler datasets and remain limited on more complex tasks.

On synthetic datasets, few-shot learning performs very well in low-difficulty settings and remains robust to structured harmonic complexity, with richer harmonic structure producing more distinctive class prototypes. The performance drop observed at intermediate harmonic levels can be attributed to increased class overlap, while strong additive noise leads to a progressive degradation of performance.

Overall, these results show that few-shot performance depends on the dataset size and complexity, with more complex datasets such as *UrbanSound8K* exhibiting lower performance.

5.2 Few-shot Learning vs Fine-Tuning

Dataset	**1-shot**	**5-shot**	**10-shot**	**FT**
Cats vs. Dogs vs. Birds	72.8	92.1	95.4	97.8
Snoring	55.00	62.8	66.8	95.3
UrbanSound8K	20.7	22.7	22.1	14.6

Table 3: Few-shot accuracy ($k \in \{1, 5, 10\}$) averaged over 100 episodes. Fine-tuning accuracy: 30 epochs.

Additionnal result are reported in Appendix B.

Table 3 compares few-shot learning with supervised fine-tuning on three datasets of increasing complexity. On the simple *Cats vs. Dogs vs. Birds* task, few-shot learning rapidly approaches fine-tuning performance as the number of support examples increases. On *Snoring Detection*, fine-tuning provides a substantial improvement over few-shot learning, reflecting the benefit of task-specific adaptation in the presence of background noise and acoustic variability. In contrast, on *UrbanSound8K*, few-shot learning slightly outperforms fine-tuning, likely due to the limited amount of labeled data and the difficulty of effectively adapting a large model to a highly diverse multi-class task.

These results suggest that, for certain tasks, few-shot learning should be preferred over supervised fine-tuning. In particular, on simpler or moderately complex datasets such as *Cats vs. Dogs vs. Birds* and *TIMIT*, few-shot learning achieves strong performance with very limited supervision, while avoiding the computational cost and instability associated with fine-tuning. Even on more challenging tasks, few-shot learning remains a competitive alternative, demonstrating that effective adaptation can be achieved with minimal labeled data.

6 Conclusion

In this work, we investigated few-shot learning as a lightweight adaptation strategy for speech foundation models, using HuBERT as a frozen feature extractor. Through benchmarking on a diverse set of real-world and synthetic audio classification datasets, we showed that few-shot learning can achieve strong performance with very limited labeled data, particularly on simple and moderately complex tasks.

Our results highlight that few-shot performance depends on dataset size and complexity. While performance generally improves with additional support examples, gains tend to saturate on simpler datasets and remain limited on more challenging tasks. Comparisons with supervised fine-tuning further reveal a trade-off between predictive accuracy and computational cost, with few-shot learning emerging as a practical alternative in resource-constrained settings.

Finally, these findings open perspectives for future work, including hybrid adaptation strategies that combine few-shot learning with parameter-efficient fine-tuning, as well as extensions to other speech foundation models and downstream tasks.

A Few-Shot Learning

A.1 Read Datasets

A.1.1 Speech Commands

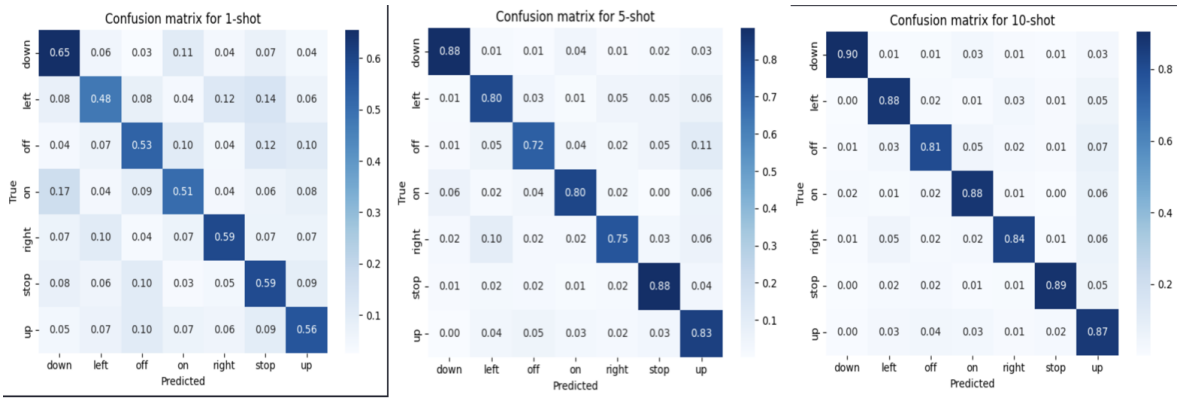

Figure 4: Confusion matrix for Speech Commands

Figure 5: t-SNE for Speech Commands

A.1.2 CREMA-D

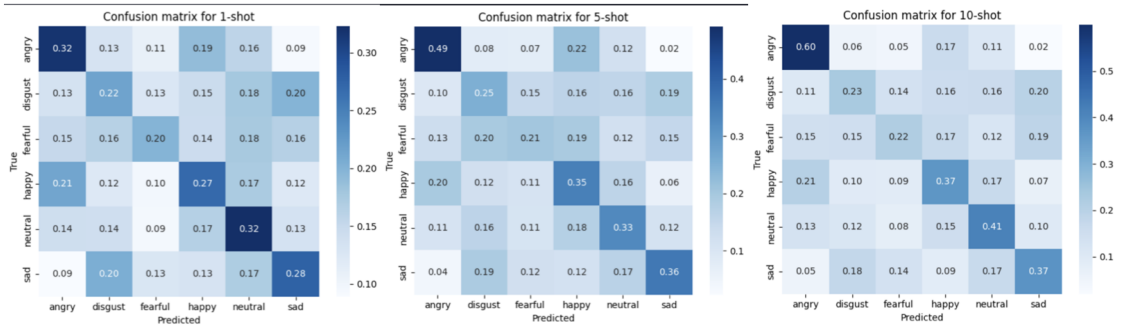

Figure 6: Confusion matrix for CREMA-D

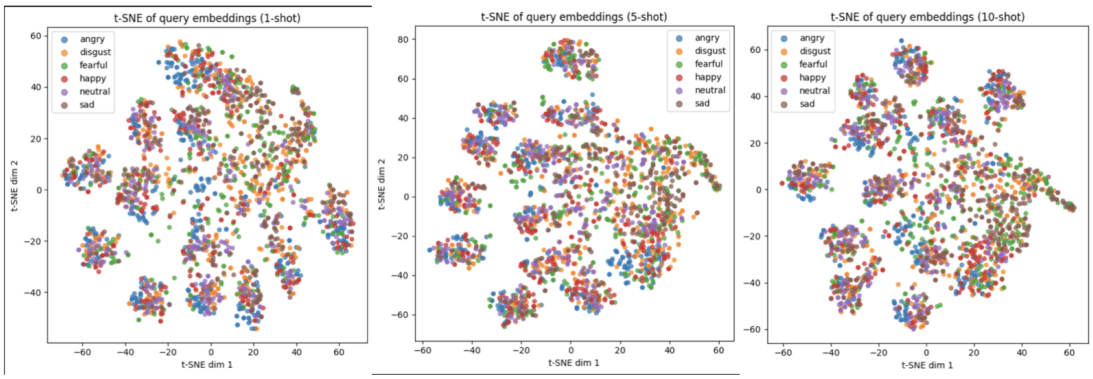

Figure 7: t-SNE for CREMA-D

A.1.3 TIMIT

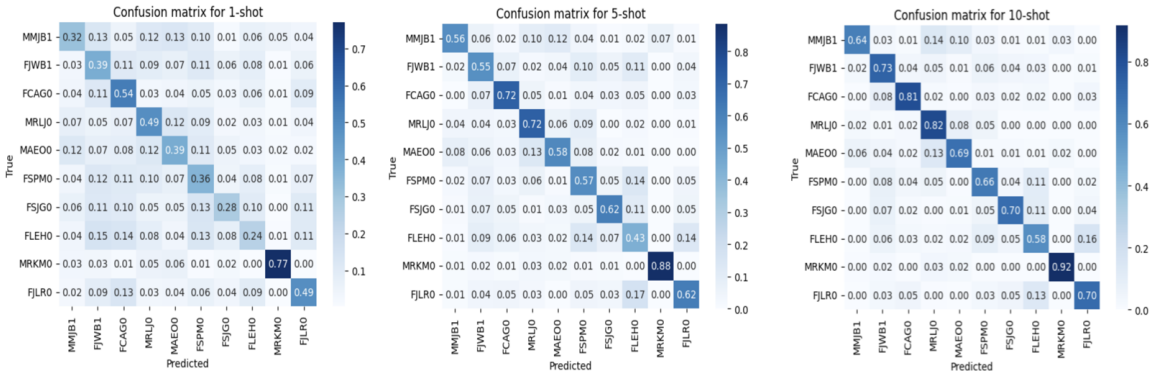

Figure 8: Confusion matrix for TIMIT

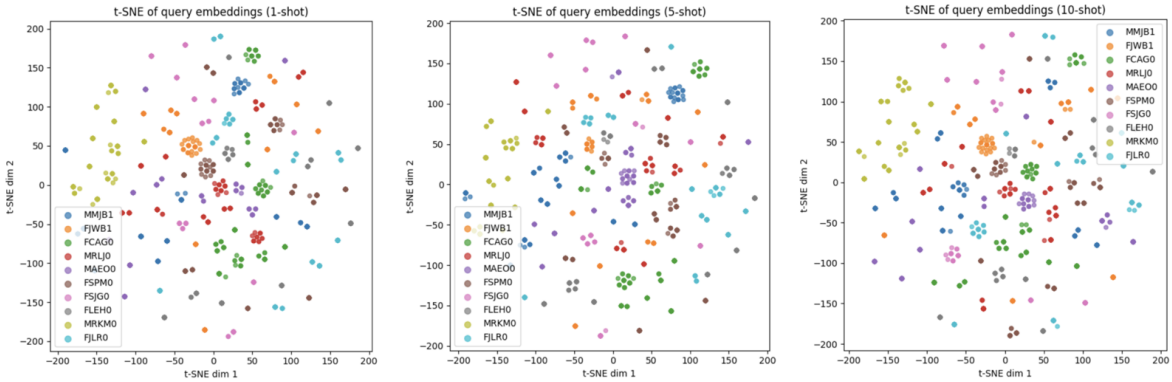

Figure 9: t-SNE for TIMIT

A.1.4 Cats vs. Dogs vs. Birds

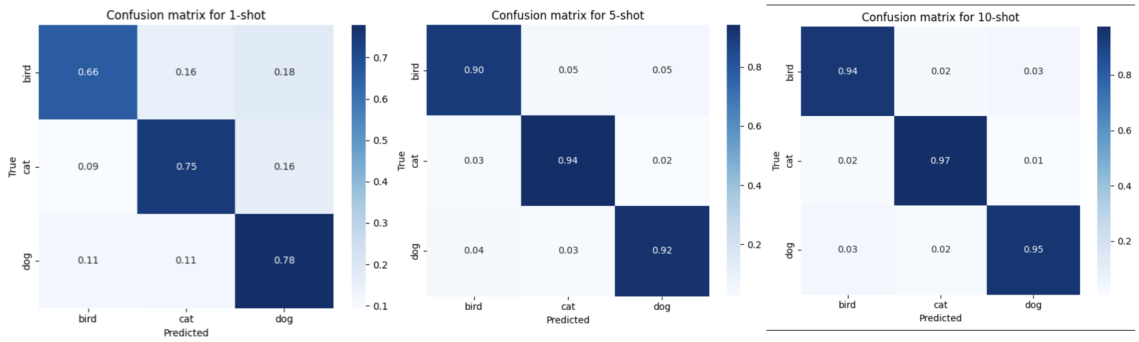

Figure 10: Confusion matrix for Cats vs. Dogs vs. Birds

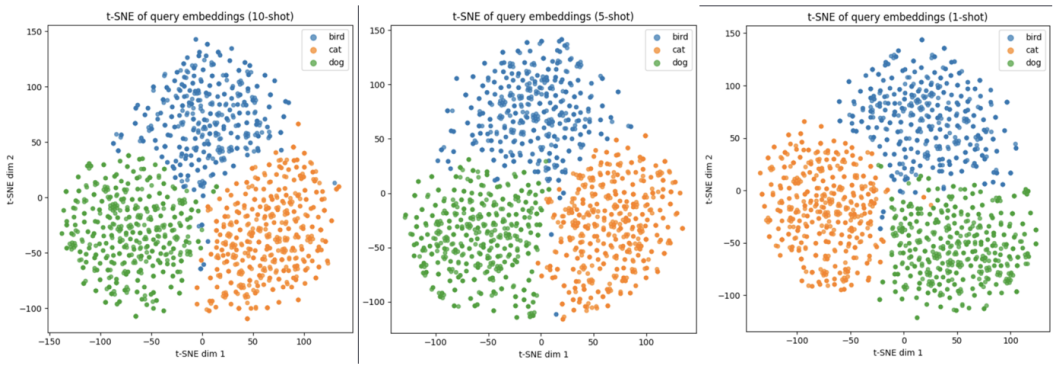

Figure 11: t-SNE for Cats vs. Dogs vs. Birds

A.1.5 Snoring Detection

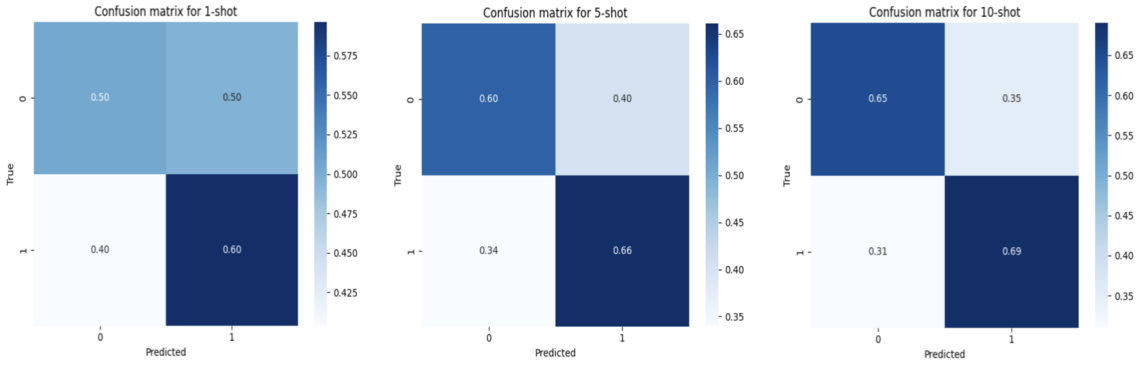

Figure 12: Confusion matrix for Snoring

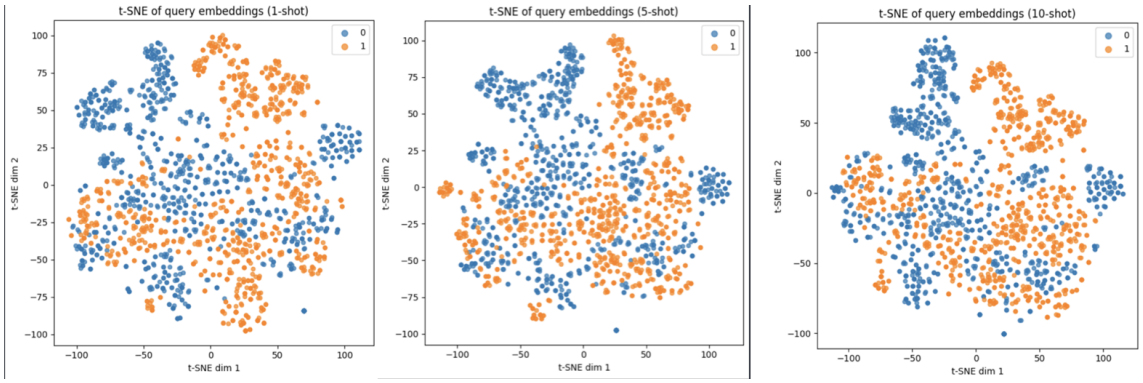

Figure 13: t-SNE for Snoring

A.1.6 UrbanSound8K

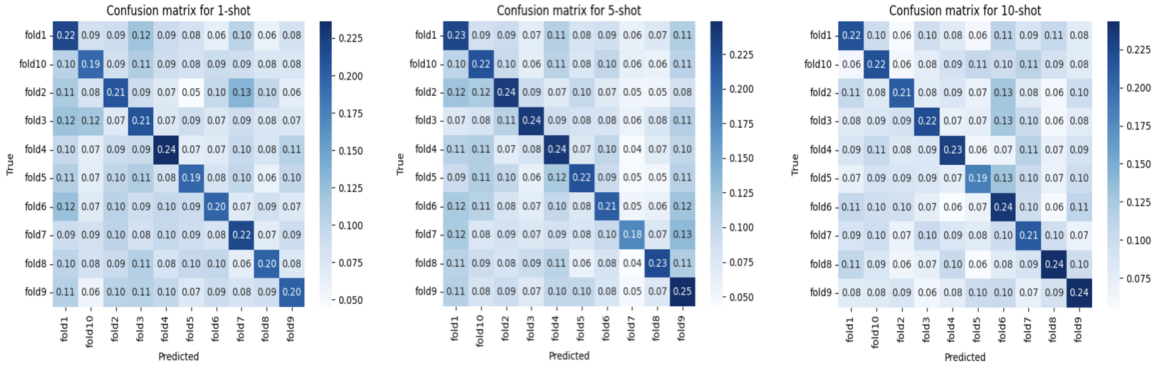

Figure 14: Confusion matrix for UrbanSound8K

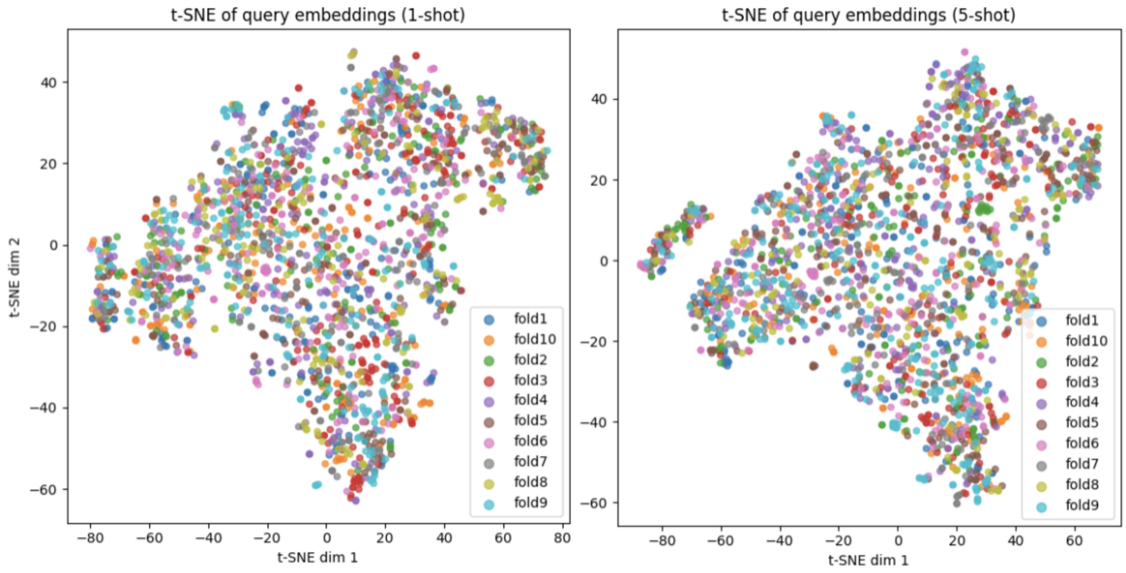

Figure 15: t-SNE for UrbanSound8K

A.2 Synthetic Datasets

A.2.1 Audio Noise Dataset (Low)

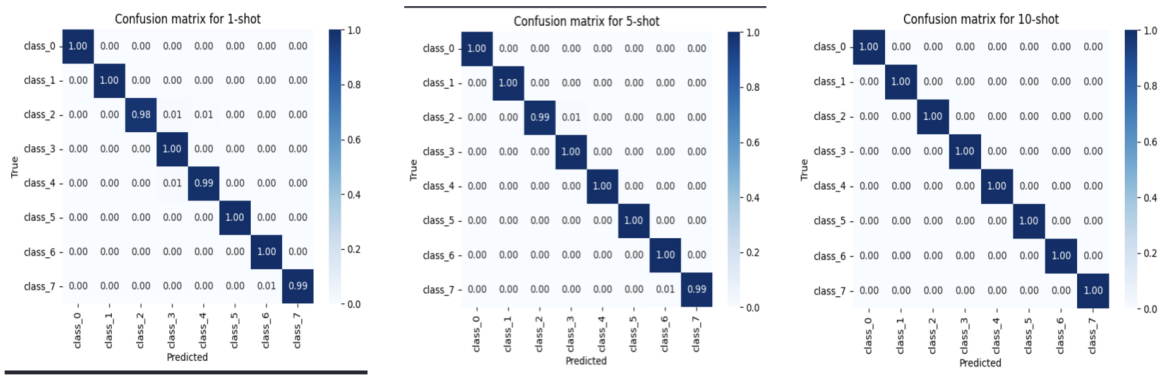

Figure 16: Confusion matrix for Synthetic Audio Noise (Low).

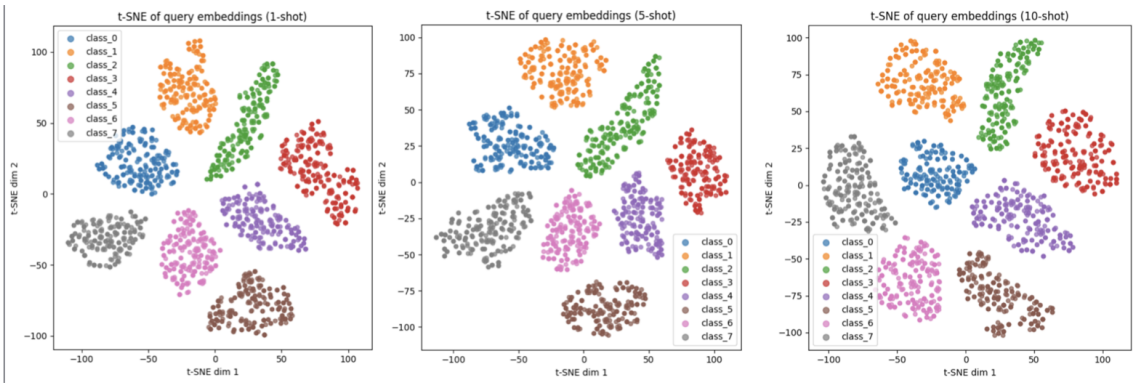

Figure 17: t-SNE visualization for Synthetic Audio Noise (Low).

A.2.2 Audio Noise Dataset (Mid)

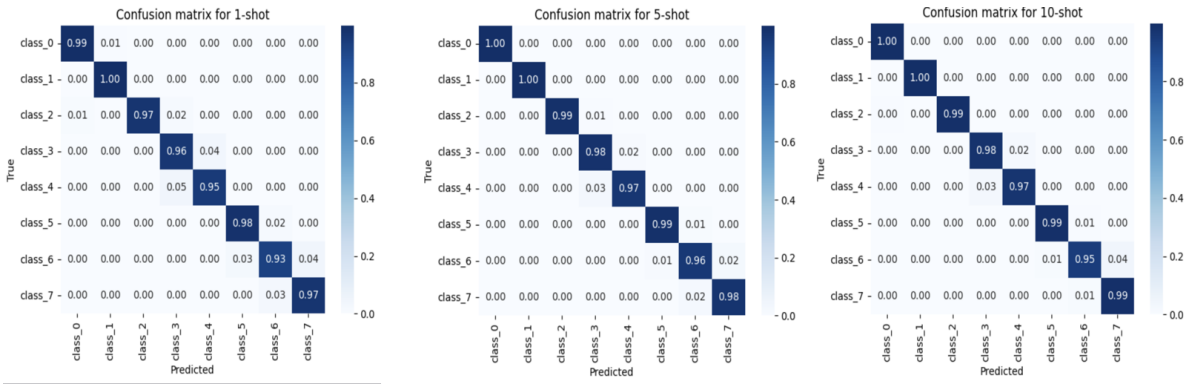

Figure 18: Confusion matrix for Synthetic Audio Noise (Mid).

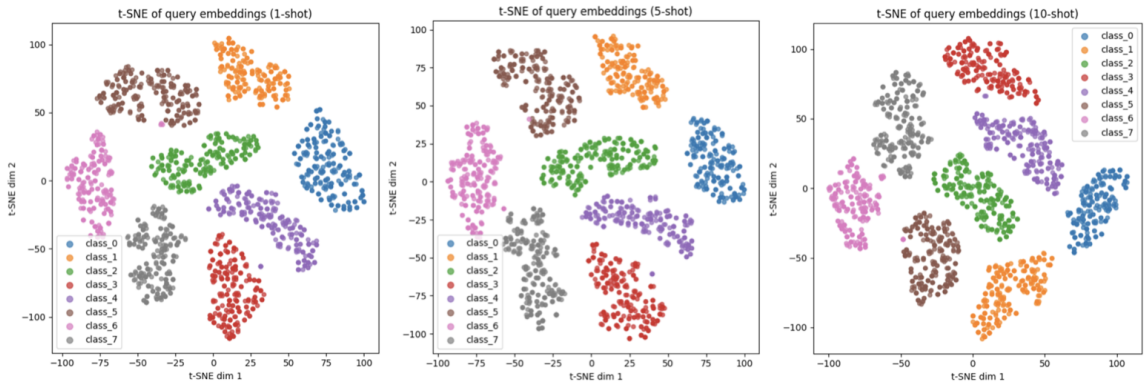

Figure 19: t-SNE visualization for Synthetic Audio Noise (Mid).

A.2.3 Audio Noise Dataset (High)

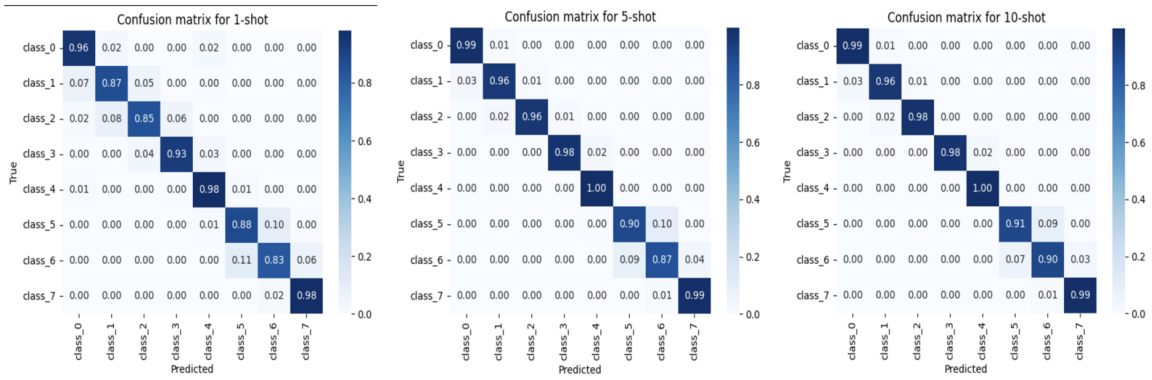

Figure 20: Confusion matrix for Synthetic Audio Noise (High).

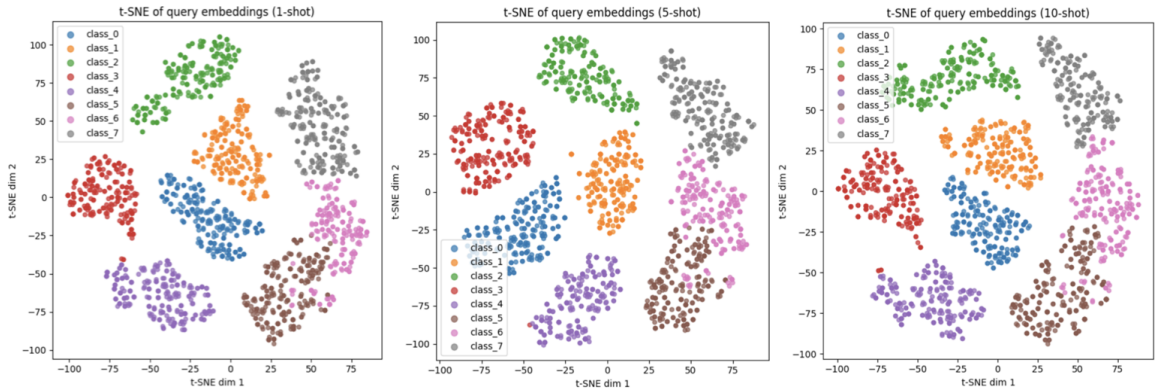

Figure 21: t-SNE visualization for Synthetic Audio Noise (High).

A.2.4 Audio Harmonics Dataset (Low)

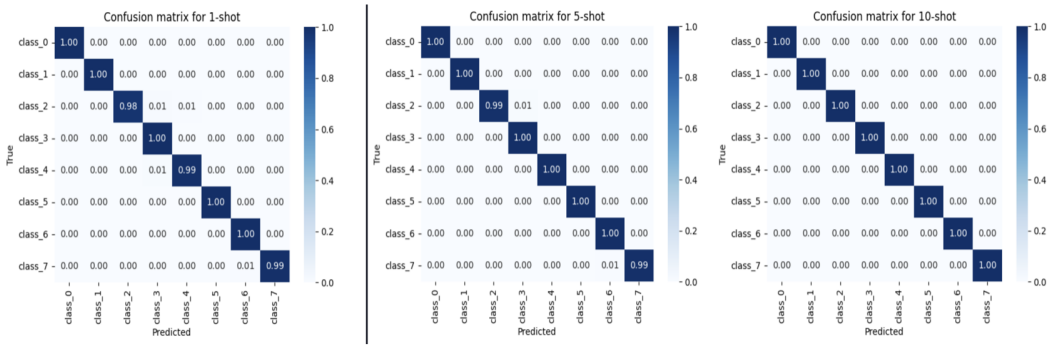

Figure 22: Confusion matrix for Synthetic Audio Harmonics (Low).

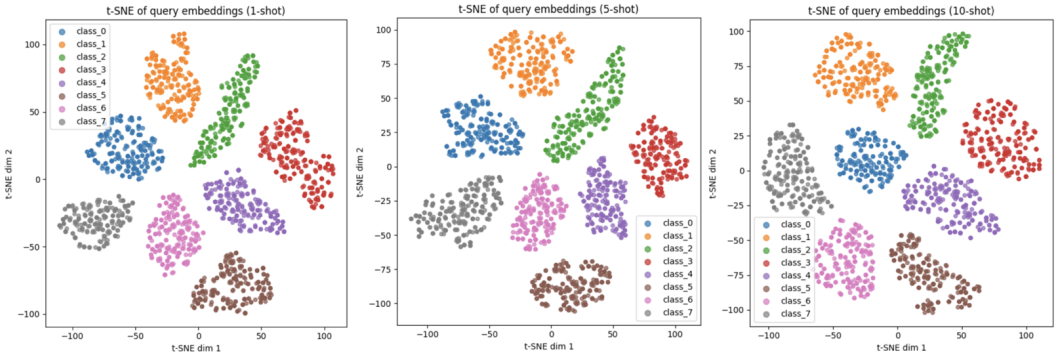

Figure 23: t-SNE visualization for Synthetic Audio Harmonics (Low).

A.2.5 Audio Harmonics Dataset (Mid)

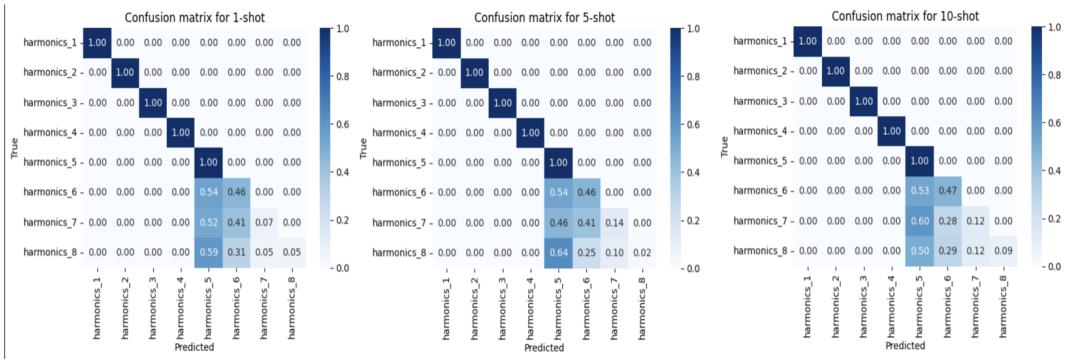

Figure 24: Confusion matrix for Synthetic Audio Harmonics (Mid).

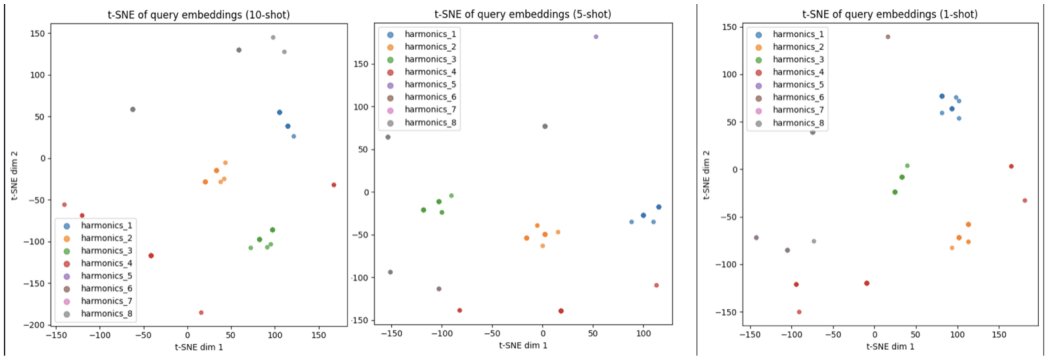

Figure 25: t-SNE visualization for Synthetic Audio Harmonics (Mid).

A.2.6 Audio Harmonics Dataset (High)

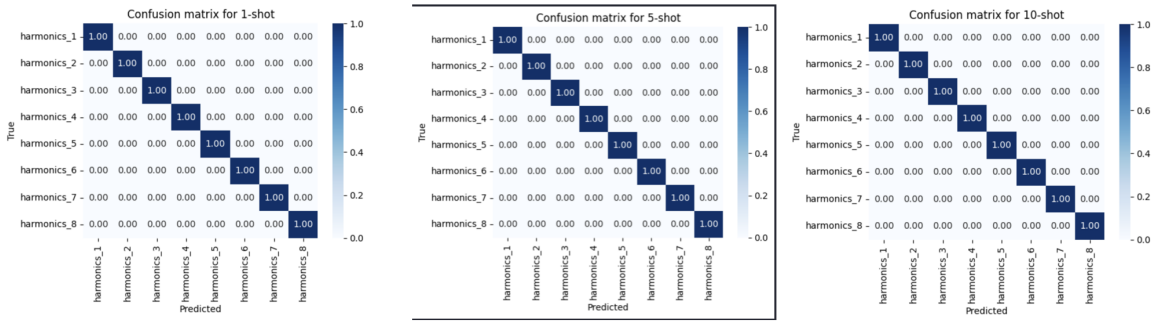

Figure 26: Confusion matrix for Synthetic Audio Harmonics (High).

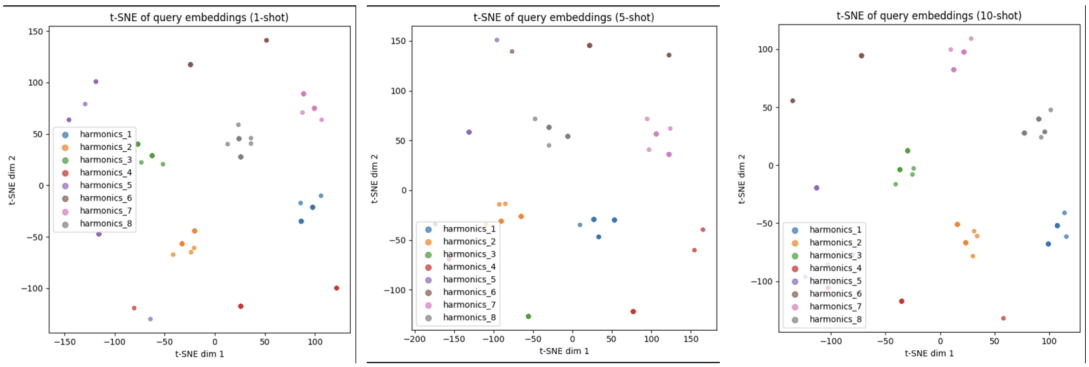

Figure 27: t-SNE visualization for Synthetic Audio Harmonics (High).

B Fine tuning

B.1 Real Datasets

B.1.1 Cats vs. Dogs vs. Birds

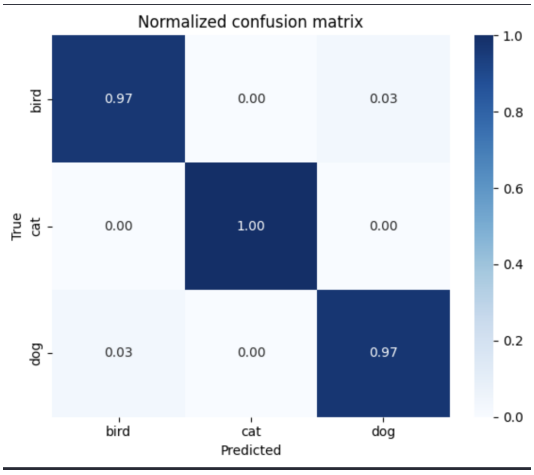

Figure 28: Confusion matrix for Cats vs. Dogs vs. Birds

B.1.2 Snoring Detection

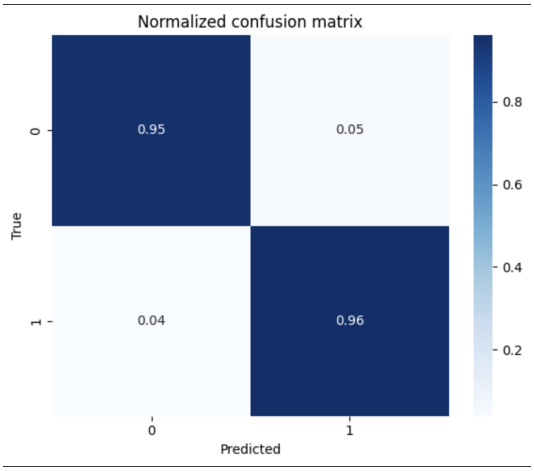

Figure 29: Confusion matrix for Snoring Detection

B.1.3 UrbanSound8K

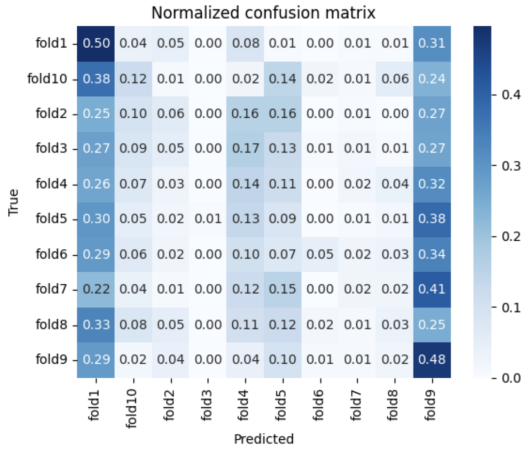

Figure 30: Confusion matrix for UrbanSound8K

References

[1] Houwei Cao, David G. Cooper, Michael K. Keutmann, et al. Crema-d: Crowd-sourced emotional multimodal actors dataset. *IEEE Transactions on Affective Computing*, 2014.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2019.

[3] John S. Garofolo, Lori F. Lamel, William M. Fisher, et al. Timit acoustic-phonetic continuous speech corpus. *Linguistic Data Consortium*, 1993.

[4] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2021.

[5] Edward J. Hu, Yelong Shen, Phil Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

[6] Justin Salamon, Christopher Jacoby, and Juan Pablo Bello. A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM International Conference on Multimedia*, 2014.

[7] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, 2017.

[8] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[9] Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.