Poster: Resource-Efficient Environmental Sound Classification Using Hyperdimensional Computing

Run Wang*
ruw041@ucsd.edu
University of California, San Diego
California, USA

Shirley Bian*
y1bian@ucsd.edu
University of California, San Diego
California, USA

Xiaofan Yu
x1yu@ucsd.edu
University of California, San Diego
California, USA

Quanling Zhao
quzhao@ucsd.edu
University of California, San Diego
California, USA

Le Zhang
lez014@ucsd.edu
University of California, San Diego
California, USA

Tajana Rosing
tajana@ucsd.edu
University of California, San Diego
California, USA

Abstract

On-device environmental sound classification (ESC) in rural areas faces one major challenge of resource efficiency. Traditional methods rely on resource-intensive machine learning models, making them impractical for small edge devices like microcontrollers (MCUs). This poster presents SoundHD, a novel ESC solution using Hyperdimensional Computing (HDC), a brain-inspired and lightweight computing paradigm. We further optimize the memory footprint for deployment on MCUs. Our initial results show that SoundHD can be deployed and executed effectively on memory-constrained MCUs.

CCS Concepts

• **Computing methodologies** → **Supervised learning by classification**; • **Computer systems organization** → **Embedded software**; • **Theory of computation** → **Models of learning**.

Keywords

Environmental Sound Classification, Hyperdimensional Computing, Embedded, Edge Device, Machine Learning

ACM Reference Format:
Run Wang*, Shirley Bian*, Xiaofan Yu, Quanling Zhao, Le Zhang, and Tajana Rosing. 2024. Poster: Resource-Efficient Environmental Sound Classification Using Hyperdimensional Computing. In *ACM Conference on Embedded Networked Sensor Systems (SenSys '24), November 4–7, 2024, Hangzhou, China.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3666025.3699427

1 Introduction

Environmental sound classification (ESC) is essential for managing biological and human environments, including wildlife monitoring and urban sound detection [6]. ESC devices are often deployed in rural areas with limited connectivity and electricity, requiring local execution on battery-powered devices. Therefore, developing resource-efficient and sustainable ESC solutions is crucial, especially for small platforms like MCUs [6]. Conventional ESC algorithms rely on complex neural networks to achieve accurate predictions. Models like Transformer-based Beats have up to 90M parameters [3]. Even the smallest, ACDNet [5] has 303kB of SRAM and requires 2.7s to perform one inference on a high-end MCU. These methods are impractical for resource-constrained MCUs, which have limited SRAM (≤ 256KB) and computational capabilities.

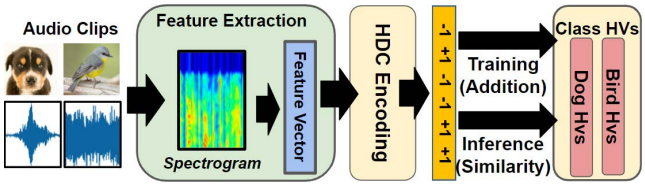

Figure 1: Overview of SoundHD including training and inference.

To overcome these limitations, we introduce SoundHD, the first ESC framework based on Hyperdimensional Computing (HDC). HDC is a brain-inspired computing paradigm that operates in a high-dimensional vector space, offering lightweight training and minimal memory requirements [8]. Consequently, SoundHD offers superior memory and computational efficiency compared to existing models, making it ideal for resource-constrained devices such as MCUs. We further propose a compression technique to reduce the memory footprint of SoundHD, allowing it to fit on MCUs. SoundHD enables lightweight on-device learning for rural sound monitoring with limited on-board resources.

2 Method and Implementation

Based on HDC, SoundHD represents sound clips in high-dimensional and low-precision vectors, referred to as hypervectors [8]. Learning is performed through simple element-wise operations on these hypervectors. In this section, we provide a detailed explanation of SoundHD's HDC-based learning process, compression techniques, and system implementation.

HDC Learning: Figure 1 provides an overview of HDC-based learning in SoundHD. We begin by extracting features from raw sound clips, which applies Mel-Spectrogram transformation and calculates frequency channel-wise statistics, such as the mean and standard deviation. This process also incorporates Mel-Frequency Cepstral Coefficients (MFCCs). We extract a total of $d = 256$ features.

*Both authors contributed equally to this research.

SenSys '24, November 4–7, 2024, Hangzhou, China

ACM ISBN 979-8-4007-0697-4/24/11
https://doi.org/10.1145/3666025.3699427

Next, we encode the sound features into hypervectors with a dimension of D. We use bipolar random projection to better utilize the on-board memory. Formally, suppose x is a sound clip and $f(\cdot)$ denotes the aforementioned feature extraction. The HDC encoding can be expressed as $\phi(X) = sign(M \times f(x))$, where M is a random matrix of shape $D \times d$ uniformly sampled from $\{-1, 1\}$. Note, that each dimension of $\phi(X)$ is bipolar (either 1 or -1) after encoding.

The main training process of SoundHD is to create class hypervectors that represent the common patterns for each sound class. This is done by combining all hypervectors from the same class via element-wise addition. For inference on an unseen sound clip, SoundHD performs a simple similarity check (i.e., cosine similarity) between the hypervector of the new clip, and all existing class hypervectors. The class with the highest similarity score indicates the predicted class label.

Hypervector Compression: Thanks to the bipolar representation of hypervectors, SoundHD can be efficiently compressed to fit smaller memory capacities. We propose a compression technique that maps matrix element 1 to a bitwise 1, and -1 to a bitwise 0. Each dimension of hypervectors is compressed and stored as one bit to reduce memory footprint. Since most MCUs use a byte-based architecture, we fit eight dimensions into one byte, achieving an 8× compression rate. We apply this compression to the projection matrix, which is the most memory-intensive component of our design. For other components, such as class hypervectors and train/test labels that do not require high precision, we downcast their data type from float to char, the smallest data type supported by C.

Implementation: Existing HDC frameworks such as torch-hd [4] are Python-based and are difficult to run on low-end MCUs based on C. Thus, we develop an embedded HDC framework specifically for memory-constrained MCUs. The overall memory usage of SoundHD in bytes can be estimated using the equation $(C + \frac{T \times P}{8} + \frac{d}{8}) \times D$, where T is the total number of clip samples, P is the proportion of train data, D is the hypervector dimension, C is the number of classes, and d is the initial dimension of the extracted features before encoding. By adjusting these parameters according to the environment, developers can easily adapt our framework to any MCUs with various memory capacities.

3 Experiments and Preliminary Results

Experimental Setup: We implement SoundHD on the Arm Cortex M4-based Arduino Nano 33 BLE board [1] with 256KB SRAM and 1MB flash memory. We use two datasets, BDLib [2] and ESC-10 [6], both sampled at 44.1 kHz. The raw audio is segmented into 0.5-second clips with 50% overlap, followed by a 70%/30% train-test split. To learn more generalized features, we augment the training audios using pitch shift and time stretch. We implement feature extraction using the Librosa library [7].

Metrics: We evaluate the accuracy of ESC, compare memory usage and inference latency on MCUs across various methods and settings.

Preliminary Results: We compare SoundHD to the state-of-the-art ESC model, ACDNet [5], on the ESC-10 [6] dataset. SoundHD saves memory usage by 4× compared to ACDNet, reducing memory requirement from 803KB to 184KB. SoundHD also significantly reduces the inference latency from 2.7 seconds to 36 milliseconds, showing a 75× improvement. This makes SoundHD ideal for real-time ESC tasks on resource-constrained devices.

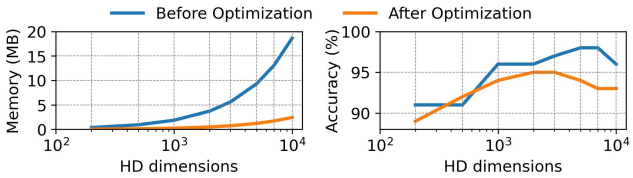

Figure 2: Memory usage (left) and accuracy (right) comparisons before and after compression for BDLib [2].

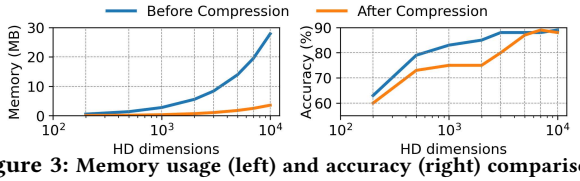

Figure 3: Memory usage (left) and accuracy (right) comparisons before and after compression for ESC-10 [6].

We evaluate the proposed compression technique by comparing the uncompressed HDC model with SoundHD, as shown in Figure 2 and Figure 3. The memory comparisons indicate a significant reduction in memory usage - approximately 7× smaller than the baseline HDC implementation - thanks to our hypervector compression algorithm. Despite smaller memory footprint, our resource-efficient implementation maintains little drops in accuracy, with less than 5% for BDLib [2] and less than 9% degradation for ESC-10 [6]. After compression, we notice SoundHD with $D = 1000$ for BDlib [2] and $D = 500$ for ESC-10 [6] fit within the target MCU with minimal accuracy degradation. Specifically, SoundHD reduces memory usage from 1864KB to 243KB on BDLib [2] with $D = 1000$, making it feasible for implementation on memory-constrained MCUs.

4 Discussion and Future Work

In this poster, we present SoundHD, a resource-efficient HDC framework for ESC. Our results show that we save 4× memory usage and 75× inference time compared to the baseline methods. SoundHD enables resource-efficient and real-time ESC on the edge. In future work, we will explore the on-device training using HDC in a dynamic and complex sound environment.

Acknowledgements

This work was supported in part by National Science Foundation under Grants #2003279, #1826967, #2100237, #2112167, #1911095, #2112665, #2120019, #2211386 and in part by PRISM and CoCoSys, centers in JUMP 2.0, an SRC program sponsored by DARPA.

References

[1] Arduino. Arduino nano 33 ble, 2024.
[2] Bountourakis, V., et al. Machine learning algorithms for environmental sound recognition: Towards soundscape semantics. In *Proceedings of the audio mostly 2015 on interaction with sound*. 2015, pp. 1–7.
[3] Chen, S., et al. Beats: Audio pre-training with acoustic tokenizers. *arXiv preprint arXiv:2212.09058* (2022).
[4] Heddes, M., et al. Torchhd: An open source python library to support research on hyperdimensional computing and vector symbolic architectures. *Journal of Machine Learning Research 24*, 255 (2023), 1–10.
[5] Mohaimenuzzaman, M., et al. Environmental sound classification on the edge: A pipeline for deep acoustic networks on extremely resource-constrained devices. *Pattern Recognition* (2022), 109025.
[6] Piczak, K. J. Esc: Dataset for environmental sound classification. In *Proceedings of the 23rd Annual ACM Conference on Multimedia*, ACM Press, pp. 1015–1018.
[7] Team, L. D. Feature extraction – librosa 0.10.2 documentation, 2023.
[8] Thomas, A., et al. A theoretical perspective on hyperdimensional computing. *Journal of Artificial Intelligence Research 72* (2021), 215–249.