A Transformer-free signal encoding module for efficient networked AI systems

Linh Vu
linh.vu@monash.edu
Monash University
Malaysia

Wern-Han Lim
lim.wern.han@monash.edu
Monash University
Malaysia

Raphaël C.-W. Phan
raphael.phan@monash.edu
Monash University
Malaysia

ABSTRACT

This research introduces a framework for constructing networked artificial intelligence systems featuring a lightweight neural network front-end tailored for long and intricate sequential data, such as audio voice recordings and health signals. Our approach uses a client-server design pattern, resulting in a compact and modular design that can be easily optimized for deployment on edge devices while still being able to incorporate more powerful backbone models. We tested the proposed blueprint on four different problem domains, including audio keyword spotting, speech emotion recognition, abnormal heart sound detection, and sentiment classification from social media text posts. The results showed an unweighted accuracy of 86%, 69%, 93%, and 95%, respectively, which are comparable or superior to other state-of-the-art methods that rely on pretrained models or pre-processing pipelines. Additionally, end-users' privacy is protected as their sensitive data are encoded and compressed before being sent over the network. These are essential aspects that machine learning practitioners should consider when designing networked AI applications in real-world scenarios.

CCS CONCEPTS

• **Computing methodologies** → **Speech recognition**; **Information extraction**; • **Applied computing** → *Bioinformatics*; • **Computer systems organization** → *Client-server architectures.*

KEYWORDS

lightweight ML, speech emotion recognition, text sentiment, AI in healthcare, signal processing

ACM Reference Format:
Linh Vu, Wern-Han Lim, and Raphaël C.-W. Phan. 2024. A Transformer-free signal encoding module for efficient networked AI systems. In *The 2nd Workshop on Networked AI Systems (NetAISys '24), June 3–7, 2024, Minato-ku, Tokyo, Japan.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3662004.3663550

NetAISys '24, June 3–7, 2024, Minato-ku, Tokyo, Japan

ACM ISBN 979-8-4007-0661-5/24/06.
https://doi.org/10.1145/3662004.3663550

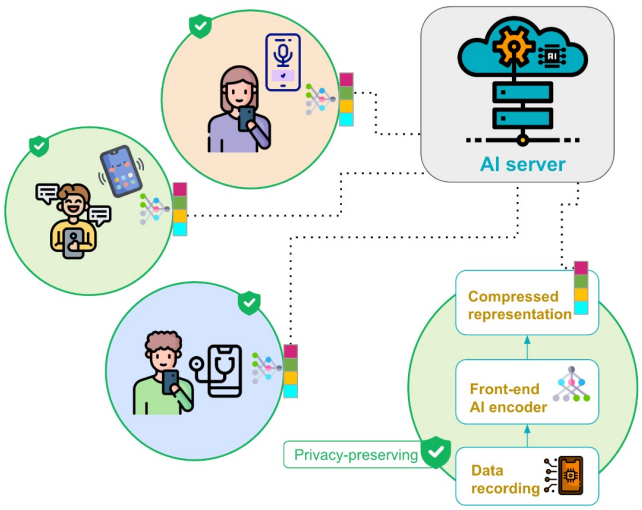

Figure 1: A high-level design for Networked AI systems

1 INTRODUCTION

In recent years, there has been a growing demand for lightweight and portable machine learning (ML) models that can run within personal devices, such as smartphones, smartwatches, or Internet of Things (IoT) devices. However, prevalent ML systems for speech recognition or natural language processing (NLP) often rely on large models such as ResNet [7, 16] or Transformer [1, 4, 5, 9, 22] that need significant computational resources and memory allocation, hence requiring cloud environments to operate. We aim to advance the field by introducing a novel front-end neural network architecture adept at processing raw heterogeneous data without the need for preprocessing or external libraries. Our approach promises versatility and efficiency in real-world applications by seamlessly accommodating diverse data types, such as audio recordings and noisy social media text posts, including emojis, hashtags, links, and uppercase letters.

The key contributions of our study include:

- Proposal of a **Transformer-free** AI architecture blueprint for on-device raw data encoding, yielding high accuracy in tasks like audio keyword spotting, speech emotion recognition, abnormal heart sound detection, and sentiment classification.
- Introduction of a **transparent** multichannel signal processing approach within the neural network, enhancing efficiency and interpretability.

- The proposed approach enables **privacy-preserving** through client-side data encoding and compression, adding another layer towards ensuring confidentiality in data transmission for networked AI systems.

The subsequent sections of this paper are outlined as follows: Section 2 illuminates the underlying motivation sourced from the field of signal processing. Next, section 3 elaborates on the system design, highlighting the novel front-end component. Section 4 introduces a series of case studies with experiment results and discussions. Finally, we conclude and sum up our findings with Section 5.

2 MOTIVATION

Sub-band coding techniques, commonly used in MPEG audio compression, utilized psychoacoustics to assign bits to perceptually significant audio signal components. However, these techniques may not always produce the most optimal representations for ML models. High-dimensional data requires complex deep learning (DL) architectures for accurate pattern identification. Furthermore, transmitting sensitive data over networks can raise privacy concerns.

Nonetheless, the building blocks of sub-band coding, such as Mel filter banks [23] or wavelet transforms [12, 20], can provide valuable insights into designing more efficient end-to-end DL models for signal data. Rather than developing separate preprocessing steps for each audio compression format, a front-end module incorporating signal processing techniques can act as a sub-band signal encoder to learn a compact data representation optimized for ML tasks. By transmitting only the compressed data over the network, this approach reduces the amount of data transferred and addresses privacy concerns associated with sensitive data types. This serves as the key motivation for the proposed system, which will be detailed in the following section 3.

3 SYSTEM DESIGN

In this section, we will discuss the overview of the proposed system design for networked AI systems before taking a detailed look at the novel front-end module for efficient on-device sequential data encoding. The back-end module will be detailed in the subsequent section, along with the experiment results.

3.1 A blueprint for networked AI systems

Figure 1 depicts a high-level architecture design of a client-server system for networked AI. The front-end module, situated on the client-side, embodies a lightweight deep neural network (DNN) inspired by sub-band coding techniques. This front-end module decomposes signal sequences into distinct target channels, projecting them into a more compact representation before conveying them to the back-end module. The back-end module, located on the server-side, may vary from a simple ML classifier or regressor to a sophisticated multitask DL model, depending on the application's requirements. Notably, the back-end module can either coexist with the front-end module on the edge device or operate within a distributed high-performing cluster, illustrating the system's adaptability.

3.2 A lightweight sequential data encoder

In digital signal processing (DSP), Firwin, or Finite-impulse response (FIR) design using the window method, is commonly used for crafting FIR filters with precise control over the filter characteristics [18]. Firwin is suitable for a wide range of applications including audio and image processing, communications, and biomedical signal analysis. It is typically available in DSP libraries such as SciPy[1] or MATLAB. This research proposes a detailed method to integrate it into deep neural networks as a convolutional layer, namely Firwin CNN layer, facilitating the development of end-to-end DL models for signal data.

The Firwin CNN layer is a convolutional layer with Firwin filters defined by a low cut-off frequency f_0, a frequency bandwidth f_δ and a window function W with p learnable parameters $\{\phi_p\}$ for each filter. Let $H = \{h_k;\ k = \overline{0, K-1}\}$ denotes the kernel of width $K > 1$ in the time domain. $H(f_0, f_\delta, \{\phi_p\})$ is parametrically modulated by a function called $F(f_0, f_\delta)$ and $W(\{\phi_p\})$ with learnable parameters. The restricted-shaped convolution kernel H is defined as follows:

$$H(f_0, f_\delta, \{\phi_p\}) = F(f_0, f_\delta) * W(\{\phi_p\}) \tag{1}$$

The filter projection W is a General Cosine Window function defining by p learnable parameters ϕ_p:

$$W(\{\phi_p\}) = \left\{ w_k;\ k = \overline{0, K-1} \,\middle|\, w_k = \sum_{i=0}^{p} (-1)^i \phi_i \cos \frac{2\pi i k}{K-1} \right\} \tag{2}$$

By incorporating this as a parametrization of the convolutional kernels, we allow the neural network to optimize these window function parameters automatically. The desired response is constructed in the frequency domain from the given set of frequencies and gains using a K-point mesh:

$$M = \left\{ m_k;\ k = \overline{0, K-1};\ \middle|\, m_k = \frac{t}{K-1} \right\} \tag{3}$$

and the *switch* activation function:

$$\sigma(x) = \tanh\left(\frac{\beta x}{|x| + \epsilon}\right) \tag{4}$$

The proposed *switch* function enables slight and slow adjustment of the initialized frequencies for further optimization.

The given frequency bands are used to interpolate the mesh to create the desired response, followed by the application of the inverse fast Fourier transform (*iFFT*), bringing the associated filter into the time domain:

$$F(f_0, f_\delta) = iFFT\Big(\sigma\big(M - |f_0|\big) * \sigma\big(1 - M + |f_0| + |f_\delta|\big)\Big) \tag{5}$$

Thus, for each kernel, there are only p parameters ϕ_p and two band parameters f_0 and f_δ to train via gradient descent. The front-end encoder module is detailed in Figure 2, comprising two multichannel signal encoder blocks and a privacy protection block. Each multichannel signal encoder block includes a Firwin CNN layer, a residual connection, a logarithmic activation, and a normalization layer using the Local Respond Normalization function [10]. To mimic the amplitude-to-decibel conversion that human hearing experiences on a logarithmic scale, we use the Natural-logarithm ReLU activation function recommended in [14]. By stacking these sequence-to-sequence encoder blocks on top of each other, we can

[1] scipy.signal.firwin2

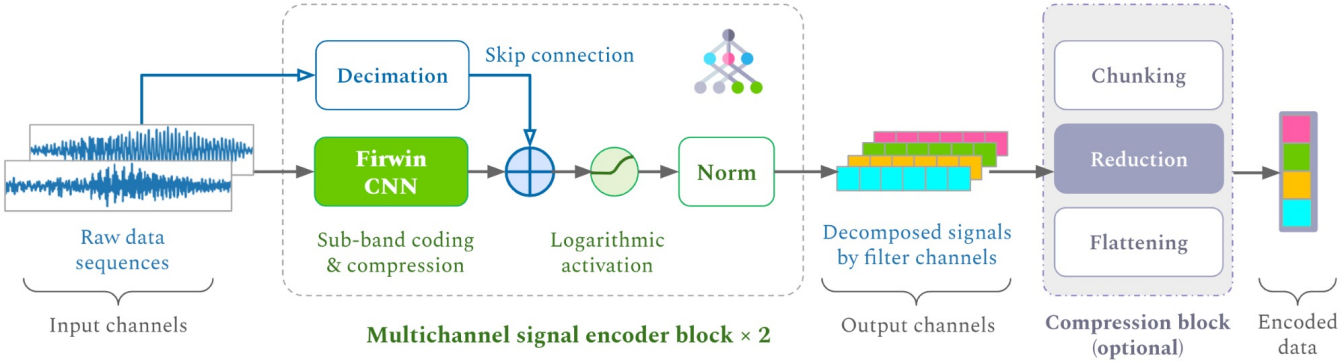

Figure 2: The proposed Front-end module, a novel CNN architecture for on-device multichannel signal encoding

achieve further pattern decorrelation from previous decomposed channels while maintaining a more compressed representation. The signal encoder blocks' outputs can be combined as long as they are resampled to the same sampling rate using decimation with a predefined *decimate_factor* to preserve the temporal characteristic. To reduce data transmission over the network and preserve end-users' privacy, the decomposed signals can be downsampled and compressed by chunking, mean or max reduction, followed by flattening (optional).

Besides the signal encoder that can be trained to optimize for the specific task, the front-end module also includes a compression block, with the design being flexible depending on the task and the backend module. For instance, in the case of audio keyword spotting and abnormal heart sound detection, a compressor utilizing mean reduction over time and returning a vector of C elements representing C sub-band channels is sufficient for identifying the spoken keyword or abnormal heart sound. For each user interaction with the system, only the vector will be transmitted over the network and can be stored on the server-side database. It is non-trivial to reconstruct the original audio from this vector representation. Therefore, our approach ensures privacy by protecting end-users' voice identity and other potentially sensitive information, such as background environment sounds that may indicate users' locations.

The front-end module can be incorporated into any deep neural network architecture, depending on the task. The subsequent section will introduce several back-end modules catering to different ML tasks with real-world applications.

4 CASE STUDIES

Our focus in this section is to investigate four distinct ML problems: audio keyword spotting, speech emotion recognition, abnormal heart sound detection, and social media text sentiment analysis. These use cases share common challenges such as handling high-dimensional and noisy data. Our proposed system effectively addresses these challenges and yields promising results across diverse domains, especially when compared to large pretrained ResNet or Transformer models.

4.1 Datasets and baselines

In speech recognition research, audio keyword spotting, a crucial element of voice-activated assistants, is the area that garners significant attention for on-device ML investigations. This section mainly summarises related works tackling this problem for on-device ML.

The Google SpeechCommands (GSC) dataset version 2, which includes 35 classes, is often used as a benchmark for evaluating different models. In [15], Rybakov et al. presented an approach to optimise models and convert them to inference mode for mobile devices. They used only 32 classes, of which 12 were main labels, and the remaining 20 were grouped into an *unknown* class. They employed data augmentation methods, Mel-frequency cepstral coefficients (MFCC) feature extraction and various CNN and RNN models such as ResNet, GRU, and Multi-headed attention RNN (MHAtt-RNN). The performance of their models on the test set, which was based on 10% of the data samples, varied between 90.6% to 98% (MHAtt-RNN), and their model sizes ranged from 354 KB to 743 KB. In [19], Soltanian et al. addressed speech command recognition in computationally constrained environments by proposing a quadratic self-organized operational layer leveraging Taylor expansion and quadratic forms, resulting in 89.8% accuracy on the 10% test set of the 10-class GSC subset using MFCC input. In the paper [16] by Shor et al., a ResNet-based model called TRILL was trained using the Triplet loss approach on Mel spectrogram input from a large YouTube dataset. Subsequently, a MobileNet [8] student model was developed to learn the latent representation from the TRILL's 19th layer. This resulted in a compressed DL model for non-semantic speech recognition tasks.

For our first experiment, **audio keyword spotting**, we utilized all 35 classes that represent 35 spoken keywords from the GSC version 2 dataset. We opted to use the raw audio recordings as the input for our model without any preprocessing steps to showcase the ability of our system to handle raw audio signals. The training set and the test set are available on *Torchaudio*[2].

In the second experiment, **speech emotion recognition** (SER), we evaluate our approach on CREMA-D [3], a popular dataset in many SER benchmarks. As human vocal expression of emotions is often vague, we focus on 4 main emotions: *Angry, Neutral, Sad,*

[2] torchaudio.datasets.SPEECHCOMMANDS

Table 1: The front-end and back-end details of proposed models and the baseline models

Dataset	Front-end module	Back-end module	Baseline model
GSC	*2 signal encoder blocks: filters=(128, 128), K=(511, 127), p=(2, 3), decimate_factor=(16, 4); compression: reduce=mean, flatten*	*Dense(dim=256)* → *LayerNorm* → *Dense(dim=256)* → *PReLU* → *Dense(dim=35)* → Output: 35 classes	[16] *Mel spectrogram* → *TRILL-MobileNet (distilled from pretrained ResNet-50)* → Output: 12 classes
Crema-D	*2 signal encoder blocks: filters=(129, 30), K=(127, 63), p=(9, 9), decimate_factor=(2, 8); compression: chunk=8, reduce=mean*	*LSTM(30)* → *Dense(dim=512)* → *LayerNorm* → *Dense(dim=512)* → *LeakyReLU* → *Dropout(0.1)* → *Dense(dim=4)* → Output: 4 classes	[16] *Mel spectrogram* → *TRILL-MobileNet (distilled from pretrained ResNet-50)* → Output: 6 classes
PASCAL	*2 signal encoder blocks: filters=(64, 64), K=(511, 127), p=(2, 9), decimate_factor=(4, 4); compression: reduce=max, flatten*	*Dense(dim=256)* → *LayerNorm* → *Dense(dim=256)* → *LeakyReLU* → *Dense(dim=2)* → Output: 2 classes	[24] *Preprocessing* → *scaled Spectrogram* → *2-layer CNN(16 filters of* 3×3*)* → *2-layer CNN(32 filters of* 3×3*)* → *Dense(128)* → Output: 2 classes
CrowdFlower	*Preprocessing: class-wise TF-IDF; 2 signal encoder blocks: filters=(128, 128), K=(511, 127), p=(2, 9), decimate_factor=(2, 8)*	*LSTM(64,64)* → *Dense(dim=256)* → *LayerNorm* → *Dense(dim=256)* → *PReLU* → *Dense(dim=13)* → Output: 13 classes	[6] *Pretrained BERTweet (with higher result than RoBERTa)* → Output: 13 classes

Happy. For these two experiments, the results from the TRILL-MobileNet model from [16] serve as the baseline.

The third application is **abnormal heart sound detection** via audio recordings using a mobile app mixed with ECG signals. The dataset was introduced in the PASCAL challenge [2] and is easily accessed via Kaggle[3]. Abnormal heart sound detection involves identifying anomalous patterns in heart sound recordings, the *murmur* heart sounds, which is essential for diagnosing cardiovascular diseases and abnormalities. The PASCAL dataset consists of two sets; one collected from the general public via a mobile app[4], and the other is from a clinic trial in hospitals using the digital stethoscope DigiScope. The main challenge of this dataset is the long duration of the recordings, which are 30 seconds on average, with mixed signals such as heart sound, breath, and background noise. Therefore, a complex preprocessing pipeline is mandatory. For example, Li et al.'s pipeline in [11] involves a 2000Hz downsampling, a 5th-order Butterworth low-pass filtering of the 0-400Hz band and the signal pre-emphasis algorithm. In the paper [24], Zhang et al. proposed 2000Hz downsampling and a 6th order Butterworth band-pass filter of range 20-950Hz and other techniques to detect heartbeat cycles and segment the audio into smaller chunks. We obtained the results from [24] for two classes: *normal* and *murmur*, from both sets, combined them with the corresponding weights based on the number of test samples to get the baseline.

Lastly, **text sentiment classification** from social media posts is a crucial NLP research area with practical applications in fields like marketing and digital health. Similar to speech emotion recognition, identifying sentiment from written text is a difficult ML task, especially from spontaneous social media text posts due to noisy content containing emojis, hashtags, links, intentionally uppercase letters and typos [21]. Guo et al. [6] conducted a benchmarking study on RoBERTa, BERTweet and ClinicalBioBERT with various social media text datasets. The results suggested that BERTweet has the ability to capture source-specific knowledge, leading to higher accuracy on many tasks. Those models were trained on vast datasets, with BERTweet trained on 80 GB and RoBERTa on 160 GB. They achieved 41.3% and 39.9% accuracies, respectively, on the CrowdFlower dataset, which is the largest evaluation set with the most emotion classes. CrowdFlower[5] is a noisy Twitter dataset consisting of 13 classes. We will demonstrate the effectiveness of our approach on this dataset versus results from Guo et al. in [6].

4.2 Proposed models & experiment setup

We used an 80:20 stratified train-test split for all experiments and no preprocess pipeline except for the CrowdFlower text dataset, where the text characters need to be transformed into numeric signals using a class-wise TF-IDF method. Table 1 describes each dataset's front-end and back-end modules. Overall, more than 60% of the model parameters, which vary between 100K and 500K, come from the back-end module. As a result, the front-end component never exceeds 500KB, satisfying the memory constraint for on-device models. We jointly trained both modules for each experiment using RAdam optimizer [13] with OneCycleLR learning rate scheduler [17] and Cross Entropy loss.

4.3 Results & discussion

The unweighted accuracy results for each classification model are reported in the table 2. Our proposed models have achieved an unweighted accuracy of 86%, 69%, 93%, and 95% on the GSC, Crema-D, PASCAL, and CrowdFlower datasets, respectively. These results are comparable or superior to other baseline methods relying on pretrained models or pre-processing pipelines.

The new front-end CNN architecture has a significant advantage in handling heterogeneous data in its raw form, such as lengthy

[3]The PASCAL dataset on *Kaggle.com*

[4]The iStethoscope Pro app for mobile heart sound recording

[5]CrowdFlower text emotion dataset on *data.world*

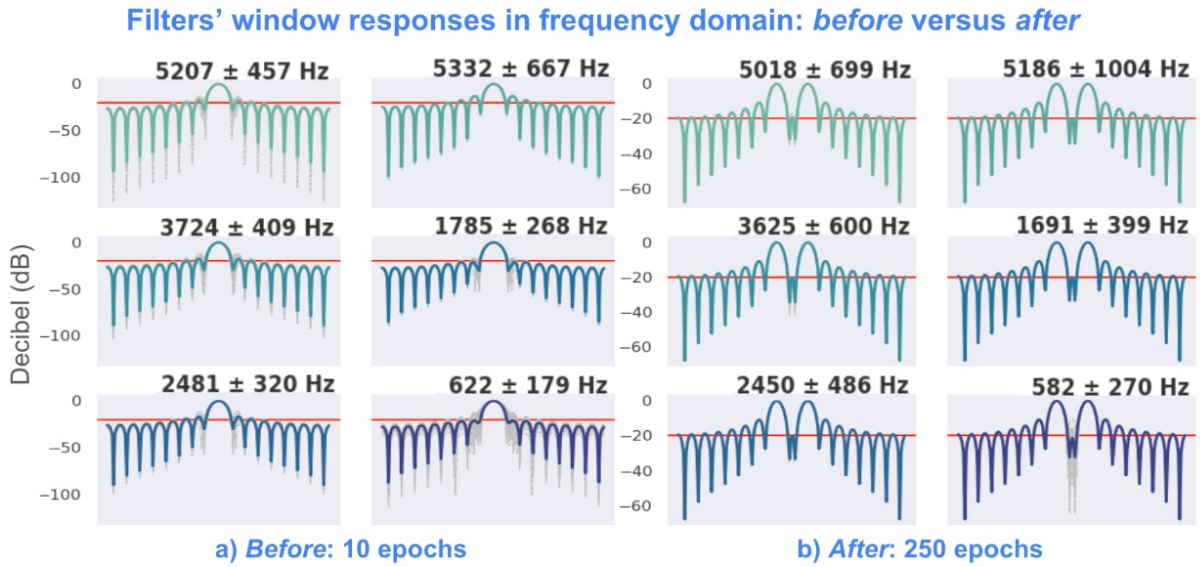

Figure 3: Filters' window responses in the frequency domain from the proposed model trained on PASCAL. Each sub-figure portrays window filters of a particular frequency range, in which the average response is in vibrant colours and all filters are in faded grey colour. The red line at -20dB represents the threshold at which noise is perceived as not noticeable.

Table 2: Evaluation of the proposed models on various ML tasks: audio keyword spotting, speech emotion recognition, abnormal heart sound detection and social media text sentiment analysis

Model	GSC	Crema-D	PASCAL	CrowdFlower
Baseline	0.75†	0.68†	0.79	0.41
Ours	**0.86**	0.69	**0.93**	**0.95**

Notation: **bold**: best results,
†: Indirect comparisons due to differences in output classes

audio recordings and noisy social media text, without requiring preprocessing or additional libraries. This versatility is particularly useful in real-world applications where data may be unstructured and varied, allowing for seamless integration and processing without the complexity of preprocessing steps. Furthermore, our proposed system design prioritizes end-users' privacy by compressing data before transmitting it over the network. This reduces the risk of privacy violations by preventing data digestion over the network or server-side data breaches.

Figure 3 demonstrates the interpretable ability of the proposed front-end module by visualizing the front-end filters after being trained on the PASCAL dataset for 10 epochs (the left figure) and 250 epochs (the right figure). In the beginning, all filters were initialized with the same band-pass filter shape. During training, the windows automatically altered their shapes to optimize signal filtering in different frequency ranges. After 250 epochs, the front-end module only allocated some band-pass filters to extract essential information in the critical range of 582 ±270 Hz. For higher frequency ranges, the windows transformed into band-stop filter shapes. Prior works in abnormal heart sound detection mentioned in the previous section 4.1 often rely on specific band-pass filtering around the same above range. This observation confirms that the front-end module functions as expected in filtering out irrelevant information for the designated ML task, which helps avoid the *garbage in - garbage out* problem. It is essential to understand the features used by the neural network model that influence its decision, especially in health applications, where reliable outcomes are paramount. Compared to other approaches using conventional CNN, Transformer or Conformer models, our proposed approach ensures transparency in neural data encoding, fostering the development of trustworthy AI models, especially for health-related applications.

5 CONCLUSION

In conclusion, the development of a lightweight neural network front-end architecture capable of processing raw heterogeneous data represents a significant advancement in the realm of networked AI systems. Our approach not only overcomes challenges associated with diverse data types and resource constraints; but also prioritizes data privacy through on-device encoding and compression. By demonstrating efficiency and transparency with smaller model sizes while maintaining comparable performance (+9%, +1%, +14%, +54% in unweighted accuracy) to established deep learning models, our method opens up new avenues for tailored deployment in real-world applications. Moreover, the potential for multimodal expansion underscores its versatility and scalability, particularly in edge computing environments. Moving forward, the integration of our proposed architecture into networked AI systems promises to enhance their adaptability, security, and efficiency.

REFERENCES

[1] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems* 33 (2020), 12449–12460.

[2] P. Bentley, G. Nordehn, M. Coimbra, and S. Mannor. [n. d.]. The PASCAL Classifying Heart Sounds Challenge 2011 (CHSC2011) Results. http://www.peterjbentley.com/heartchallenge/index.html.

[3] Houwei Cao, David G Cooper, Michael K Keutmann, Ruben C Gur, Ani Nenkova, and Ragini Verma. 2014. CREMA-D: Crowd-sourced emotional multimodal actors dataset. *IEEE transactions on affective computing* 5, 4 (2014), 377–390.

[4] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (2019).

[5] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. 2020. Conformer: Convolution-augmented transformer for speech recognition. *Proc. Interspeech 2020* (2020).

[6] Yuting Guo, Xiangjue Dong, Mohammed Ali Al-Garadi, Abeed Sarker, Cecile Paris, and Diego Mollá Aliod. 2020. Benchmarking of transformer-based pre-trained models on social media text classification datasets. In *Proceedings of the the 18th annual workshop of the australasian language technology association*. 86–91.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proc. CVPR*. IEEE, 770–778.

[8] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* (2017).

[9] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, and et al. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 29 (2021), 3451–3460.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* 25 (2012).

[11] Feng Li, Zheng Zhang, Lingling Wang, and Wei Liu. 2022. Heart sound classification based on improved mel-frequency spectral coefficients and deep residual learning. *Frontiers in Physiology* 13 (2022), 1084420.

[12] Chien-Chang Lin, Shi-Huang Chen, Trieu-Kien Truong, and Yukon Chang. 2005. Audio classification and categorization based on wavelets and support vector machine. *IEEE Transactions on Speech and Audio Processing* 13, 5 (2005), 644–651.

[13] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. 2020. On the variance of the adaptive learning rate and beyond. *Proc. ICLR 2020* (2020).

[14] Yang Liu, Jianpeng Zhang, Chao Gao, Jinghua Qu, and Lixin Ji. 2019. Natural-logarithm-rectified activation function in convolutional neural networks. In *2019 IEEE 5th International Conference on Computer and Communications (ICCC)*. IEEE, 2000–2008.

[15] Oleg Rybakov, Natasha Kononenko, Niranjan Subrahmanya, Mirkó Visontai, and Stella Laurenzo. 2020. Streaming keyword spotting on mobile devices. *Proc. Interspeech 20202* (2020).

[16] Joel Shor, Aren Jansen, Ronnie Maor, Oran Lang, Omry Tuval, Félix De Chaumont Quitry, Marco Tagliasacchi, Ira Shavitt, Dotan Emanuel, Yinnon Haviv, and et al. 2020. Towards Learning a Universal Non-Semantic Representation of Speech. *Interspeech 2020* (2020).

[17] Leslie N Smith and Nicholay Topin. 2019. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, Vol. 11006. SPIE, 369–386.

[18] Steven Smith. 2003. *Digital signal processing: a practical guide for engineers and scientists*. Newnes.

[19] Mohammad Soltanian, Junaid Malik, Jenni Raitoharju, Alexandros Iosifidis, Serkan Kiranyaz, and Moncef Gabbouj. 2021. Speech command recognition in computationally constrained environments with a quadratic self-organized operational layer. In *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–6.

[20] Pramila Srinivasan and Leah H Jamieson. 1998. High-quality audio compression using an adaptive wavelet packet decomposition and psychoacoustic modeling. *IEEE Transactions on Signal Processing* 46, 4 (1998), 1085–1093.

[21] Heng Ee Tay, Mei Kuan Lim, and Chun Yong Chong. 2022. SERCNN: Stacked Embedding Recurrent Convolutional Neural Network in Detecting Depression on Twitter. In *Proc. ICPR*. Springer.

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017), 5998–6008.

[23] Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying Liu, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, et al. 2002. The HTK book. *Cambridge university engineering department* 3, 175 (2002), 12.

[24] Wenjie Zhang and Jiqing Han. 2017. Towards heart sound classification without segmentation using convolutional neural network. In *2017 computing in Cardiology (CinC)*. IEEE, 1–4.