INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Masking Kernel for Learning Energy-Efficient Representations for Speaker Recognition and Mobile Health

Apiwat Ditthapron, Emmanuel O. Agu, Adam C. Lammert

Worcester Polytechnic Institute, 100 Institute Rd, Worcester, MA 01609 ,USA

aditthapron@wpi.edu, emmanuel@wpi.edu, alammert@wpi.edu

Abstract

Modern smartphones possess hardware for audio acquisition and to perform speech processing tasks such as speaker recognition and health assessment. However, energy consumption remains a concern, especially for resource-intensive DNNs. Prior work has improved the DNN energy efficiency by utilizing a compact model or reducing the dimensions of speech features. Both approaches reduced energy consumption during DNN inference but not during speech acquisition. This paper proposes using a masking kernel integrated into gradient descent during DNN training to learn the most energy-efficient speech length and sampling rate for windowing, a common step for sample construction. To determine the most energy-optimal parameters, a masking function with non-zero derivatives was combined with a low-pass filter. The proposed approach minimizes the energy consumption of both data collection and inference by 57%, and is competitive with speaker recognition and traumatic brain injury detection baselines.

Index Terms: windowing, energy efficiency, deep learning, speaker recognition, TBI detection

1. Introduction

Speech processing hardware embedded into smartphones facilitates on-device performance of speech tasks such as voice authentication and health assessment, either as short episodic sessions or continuously. Most state-of-the-art speech processing pipelines utilize Deep Neural Networks (DNNs) to achieve accurate analyses, with inference typically done either locally on the mobile device or on a remote server. On-device DNN inference preserves the speaker's privacy more than remote inference but requires audio to be transmitted to the server, which consumes additional energy.

Although smartphones are now powerful enough to perform real-time DNN inference, energy consumption remains an issue especially when high-performance DNNs are used for continuous, passive health assessment [1]. Prior mobile speech processing proposed improving energy efficiency by using compact DNN models [2], or energy-efficient hardware [3, 4] and feature extraction approaches [5]. DNN model complexity can be reduced by up to 86% using audio features such as Mel-frequency cepstral coefficients (MFCCs), or by factorizing the DNN model into smaller kernels using depth-wise convolution [2]. Windowing is a common speech processing step, in which the input signal is divided into temporal segments for feature extraction and DNN. Inspired by the feature compression approach, in this paper, we propose reducing DNN input dimensions by identifying the smallest usable window length and sampling rate, which in turn reduces the energy consumed by audio data acquisition and DNN inference. Our proposed method is a signal pre-processing step that can be integrated into most speech processing pipelines including compact speaker recognition and health assessment DNNs.

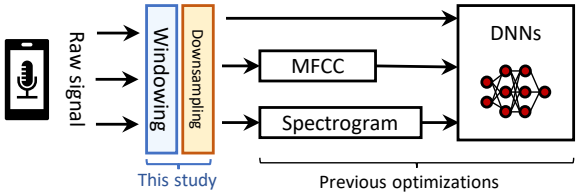

Figure 1: *Common speech processing pipeline. This study introduces learnable masking into windowing and downsampling to reduce computational complexity in downstream processing. The method can be used with speech features and DNNs previously proposed for resource-constrained computing.*

As illustrated in Figure 1, DNNs for speech processing typically operate on a sequence of discrete signal *chunks*, which are generated during pre-processing steps performed before feature extraction. Each chunk contains $n \times s$ frames from n seconds of audio sampled at s Hz and has a length that varies depending on the speech task (e.g., 200 milliseconds (ms) for speech recognition and up to 15 seconds for depression detection [6, 7]). While recording and processing longer speech chunks sampled at higher sampling frequencies improve recognition performance [4], this approach also consumes higher energy, which limits its practical application [8]. The optimal size of an audio chunk is typically determined as a hyperparameter using grid search or Bayesian optimization during DNN model training [4, 9].

To optimize the shape of the input signal and derive energy-efficient parameters, we propose determining the most energy-efficient speech duration m and sampling rate s via masking during DNN training while also learning the parameter θ, where θ are weights in the backbone DNNs. Additional windowing and down-sampling layers for optimizing m and s, as well as θ are proposed to be included at the beginning of the DNN. We envision that such parameter learning will be done during training on a remote server with inference running locally on a mobile device. Gaussian, Hamming, Hann, and Tukey windows [10] were evaluated as masking functions during back-propagation in order to learn m in the windowing layer. In contrast with the traditional use of masking (or soft-masking), we applied a binary hard mask step for constructing a discrete window. This binary step is used in the down-sampling layer as a

This material is based on research funded by DARPA under agreement number FA8750-18-2-0077.

10.21437/Interspeech.2023-1026

masking function for learning the appropriate discrete Fourier transform signal bandwidth.

Our approach is inspired by prior learning approaches that discover an optimal end-to-end DNN architecture, such as Neural Architecture Search (NAS) [11, 12]. Utilizing only training data, NAS transforms each layer of the DNN architecture into derivable functions, such as masking, which can be back-propagated via gradient descent. Flexconv [11] proposed learning the optimal kernel size for the image recognition task by using a Gaussian function as a mask on the convolution weights. DiffStride [12] proposed masking for back-propagation in order to learn the scaling factor of the pooling layer. Searching for optimal architectures using NAS achieves performance superior to hyper-parameter tuning. The use of masking in previous work is similar to ours, but the main distinctions are in the learning objective and methodology. Our method applies masking to the input, as a signal pre-processing, and not on the weights to optimize the model architecture as in NAS.

An energy-efficient penalty is introduced to prevent m and s from expanding, reducing the amount of energy required for inference and data recording on a linear scale [8]. Our proposed method is able to reduce the energy utilized for inference while minimizing losses in performance. We evaluated the windowing layer, down-sampling layer, and energy-efficient penalty at the window level (one speech chunk) and sentence level (multiple speech chunks) for the speaker recognition task, and for the continuous Traumatic Brain Injury (TBI) detection task, which are the common tasks in mobile health. The energy used for DNN inference significantly improved in all three scenarios, whereas energy expenditure during data acquisition was reduced only in the first scenario. We also show that the parameters learned in the windowing layer are compatible with the compact DNN model and compressed features and outperform the parameters obtained from traditional hyperparameter tuning, improving both accuracy and power efficiency.

2. Proposed method

The optimization of window size and the sampling rate is accomplished via back-propagation through windowing $\mathcal{W}_m$ and down-sampling layers $\mathcal{D}_s$. The parameters in these two layers (m, s) are learned jointly with the (θ) parameter in the DNN but are controlled by an energy-efficiency penalty $\mathcal{J}$ in order to minimize the size of the speech sample. Given a speech model $\mathcal{F}_\theta(x)$ with a loss function of $\mathcal{L}(x, y)$, parameters are optimized from dataset $\{x_i, y_i\}_{i=0}^{P}$ by $\text{argmin}_{\theta,m,s} \sum_{i=0}^{P} \mathcal{L}\big(\mathcal{F}_\theta(\mathcal{W}_m(\mathcal{D}_s(x_i)), y_i\big) + \mathcal{J}(m, s)$

Windowing layer: Let $x_i \in \mathbb{R}^N$ be a speech sample, composed of N frames where N is the upper bound of window length. Windowing allows a signal of length m ($\lfloor \frac{N-m}{2} \rfloor \leq n \leq \lfloor \frac{N+m}{2} \rfloor$) into the DNN. To learn m via gradient descent, derivatives of the masking function must be non-zeros. A rectangular window, a standard method for segmenting the signal for DNNs, is defined to have values of 1 within length m and values of 0 everywhere else, resulting in zero derivatives. This study proposes hard-masking, a learnable rectangle window, that uses functions with a peak at its center during back-propagation. Masking functions considered are the well-known Gaussian, Hamming, Hann and Tukey functions [10].

The Gaussian window function has a mean value of $\lfloor \frac{N-1}{2} \rfloor$ with a learnable variance σ^2. This study defines σ^2 in terms of window m at which the function is approaching zero ($m^2 = -8\log(\epsilon)\sigma^2, \epsilon = 1e^{-5}$) as $w_G(n; m) = exp(4\log(\epsilon)(n - \lfloor \frac{N-1}{2} \rfloor)^2/m^2)$. Hamming and Hann windows are defined as $w_{HM}(n; m) = 0.54 - 0.46\cos(2\pi(n - \lfloor \frac{N-m}{2} \rfloor)/(m-1))$ and $w_{HN}(n; m) = 0.5 - 0.5\cos(2\pi(n - \lfloor \frac{N-m}{2} \rfloor)/(m-1))$, respectively. Tukey is also included as a tapered cosine function of w_{HN}.

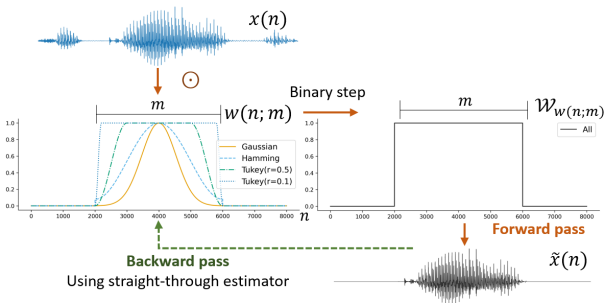

Figure 2: *Windowing layer using hard-masking*

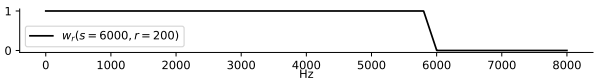

Figure 3: *Masking w_r in down-sampling layer*

A window $w(n; m)$ is applied to $x(n)$ to attenuate values outside the window. We call the output of this operation soft-masking and consider it a baseline evaluation method. To create hard-masking $\mathcal{W}_m$, a value of 1 is assigned to non-zero values of soft-masking. The hard-masking derivative of $\delta\mathcal{L}/\delta m$ is computed by applying a straight-through estimator [13] to w.

Down-sampling layer: The down-sampling layer $\mathcal{D}_s$ applies masking in the frequency domain, resampling x to $2s$ Hz. The discrete Fourier transform $X(\hat{n}) = FFT(x(n))$ is obtained from the Fast Fourier Transform (FFT) of $x(n)$. Due to Hermitian symmetry, the term with negative frequency can be disregarded. A rectangle mask is used as a low-pass filter to zero frequency bins higher than s. To reduce artifacts from the rectangle mask and allow back-propagation, a linear function is applied, which extends the cutoff frequency by r. The mask $w_r(n; s, r)$ is defined as $min(1, max(-\frac{n-s}{r})), 0 \leq n \leq N$,visualized in Figure 3. After applying $w_r(n; s, r)$ to $X(\hat{n})$, $x(n)$ is downsampled to s Hz using inverse FFT only on DFT bins between 0 and s Hz, mathematically explained by $\mathcal{D}_s = iFFT(X(\hat{n}) \odot w_r(\hat{n}; s, r))$, where $0 < \hat{n} \leq s$.

Energy-efficient penalty: A penalty term is introduced to minimize window length and sampling rate, which, in turn, reduces the energy required for data acquisition and inference. The energy-efficient penalty $\mathcal{J}(m, s) = \lambda\lceil \frac{\max(m-\mu_m, 0)}{\mu_m} + \frac{\max(s-\mu_s, 0)}{\mu_s} \rceil \bar{\mathcal{L}}$ is incorporated into the loss function to penalize $\mathcal{L}$ if m or s increases from their average values (μ_m,μ_s) in the preceding epoch. The penalty values are normalized and added proportionally to the value of $\bar{\mathcal{L}}$ (no gradient). λ is adjustable to control the penalty term. $\mathcal{J}$ is clipped at zero to prevent an exploding gradient.

3. Evaluation

The proposed method was evaluated using state-of-the-art DNNs previously proposed for speaker recognition (short) and TBI detection (continuous, passive health assessment) tasks. Our implementation is publicly available at `https:`

//github.com/aditthapron/windowMasking.

3.1. Speaker recognition task

The speaker recognition speech processing task tries to identify a speaker based on their voice characteristics. On smartphones, speaker recognition is frequently performed as continuous authentication, consuming significant energy [14].

Dataset: Text-independent speech from the TIMIT corpus was used to train and evaluate the model [15]. Read speech in English was collected from 462 speakers in 16-bits with a sampling rate of 16 kHz. All data pre-processing steps, including removing non-speech segments, removing calibration sentences, and normalizing the amplitude, were performed similarly to [6]. The space between each window center was fixed at 10ms. M was set to 500ms. The split between training and testing was the same as in [6].

Evaluation Metric: Classification Error Rate (CER) is reported at both the window and sentence levels. At the window level, the speaker with the highest negative log-likelihood is predicted, whereas, the negative log-likelihood from all windows is summed to make the prediction at the sentence level. Reduction of m in window-level speaker recognition implies a reduction in the duration of speech necessary to collect. To assess training consistency, all evaluations were repeated ten times with random seeds of varying values.

DNN architecture and features: The proposed method was evaluated for speaker recognition tasks using two DNNs, SincNet [6] and Am-MobileNet [2]. Raw audio was input to both models, whereas MFCC features were input to Am-MobileNet. SincNet [6] replaced traditional convolutional weights with the Sinc function as the kernel in CNN layers. The model consists of one CNN layer with Sinc filters and two conventional CNN layers. After the CNN, the tensor is transformed into a one-dimensional tensor to classify the speaker. SincNet can only apply to raw audio because of the Sinc layer. Am-MobileNet [2] adapted the MobileNetV2 model [16], which uses depthwise convolution and an inverted Residual Block to improve model efficiency, for the speaker recognition task. The Additive Margin (AM) was also introduced into the Softmax activation function to improve the separation margin between the decision boundary of the speaker class. For MFCC features, the first 40 Mel bands were extracted using Librosa [17]. Due to the short duration of the signal, the length of the FFT window was reduced to 1024 and the hop length to 128. Thirteen MFCCs were then extracted from Mel-spectrograms. Results with delta MFCCs were not reported as there was no performance gain.

Experiment: We extended SincNet [6] to learn energy-efficient parameters by including windowing and $\mathcal{D}_s$ layers prior to the SincNet layers. As the input shape changes throughout the learning process, the layers following the CNN were modified to only apply weights to the signal's valid length.

3.2. TBI detection task

Frequently, impaired speech is considered a TBI biomarker that can be detected via continous speech assessment using a DNN running on a smartphone, preventing fatalities and facilitating the recovery of TBI [1].

Dataset: Speech from the Coelho TBI corpus [18], was used for evaluation. The Coelho corpus contains speech during story retelling, story generation, and conversation discourses from 55 subjects with non-penetrating head injuries and 52 subjects without head injuries. We used the speech collected during the conversation discourses for evaluation. Pre-processing steps from [1], included 1) removing noisy audio, 2) normalizing audio magnitude, and 3) vocal-tract length normalization. The speech was recorded at a sampling rate of 44.1 kHz, where [1] down-sampled the signal to 16 kHz. This study initialized s in the down-sampling layer to 22 kHz.

Metric: Balanced Accuracy ,$(\text{Sensitivity} + \text{Specificity})/2$, is reported using subject-level split 10-fold cross-validation.

DNN architecture: The cascading Gated Recurrent Unit (cGRU) previously proposed for TBI detection from speech [1] was the DNN model. cGRU is a two-step DNN where the first model extracts TBI features from 200 ms of speech using five CNN layers, and the second model applies a GRU on stacked features from the first model for binary TBI classification. The CNNs were applied on 200 ms with an interval of 25 ms, and the GRU makes a TBI prediction over 4 s of speech.

Experiment: We integrated the proposed method into the cascading Gated Recurrent Unit (cGRU) model [1] to learn an energy-efficient input for the TBI detection task. Windowing and down-sampling layers were applied at the instance level ($M = 8$ s). Due to space constraints, ablation results are only reported for speaker recognition.

3.3. Model optimization Baselines and metrics

Grid search and Bayesian model-based optimization were commonly used to tune hyperparameters, including windowing parameters [19]. In this study, optimal m and s were searched in ranges of [100,300] ms and [6000,8000] Hz for speaker recognition, and [2,8] s and [6000,22050] Hz for TBI detection using grid-search. Ten evenly spaced values were used for each parameter interval. In Bayesian model-based optimization, the Tree-structured Parzen Estimator (TPE) [20] was utilized. TPE uses past evaluations of hyperparameters to construct a probabilistic model over multiple iterations. We also considered a low-pass filter with a learnable Sinc filter [6] as a baseline for $\mathcal{D}_s$ layer.

Energy-efficient metrics: The numbers of parameters and Multiply-Accumulates (MACs) have previously been shown to be effective estimations of DNN energy consumption at inference [21]. Our proposed method adds window length parameters m and sampling rate s to the model, but MACs change significantly depending on the size of the input ($m \times s$). Energy consumption at inference can be reduced by lowering the MAC, whereas power consumption during speech recording can be reduced by lowering s. Average inference time, measured on the Samsung Galaxy S22 device, and average training time, measured on the Nvidia Tesla V100, are reported.

4. Result

4.1. Speaker recognition

Windowing functions are compared as hard-masking and soft-masking for speaker recognition in Figure 4 (top). Only the hard-mask is able to optimize m, approximately at 120-200 ms. Window-CER is significantly lower than the baseline trained on a 200-ms speech. From the plot between CER and window length (Figure 5), Hamming window was the most efficient at reducing window length while maintaining the same error range as Gaussian and Tukey windows. The CER of the Hann window is lower than the other windows, however, using m higher than 200ms. Figure 4 (bottom) compares the $\mathcal{D}_s$ layer to the Sinc. $\mathcal{D}_s$ is competitive with the Sinc filter but with a significantly lower sampling rate of 7.2 kHz. Together, the two proposed masking layers can be optimized using penalty terms, as shown

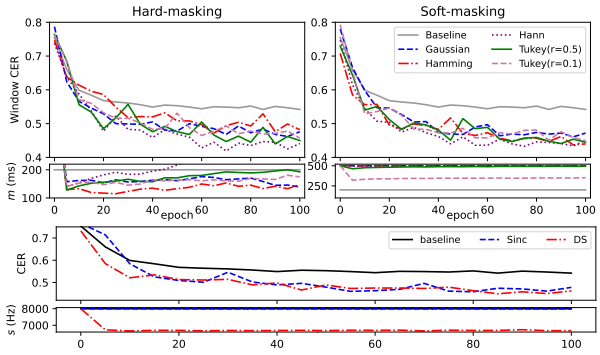

Figure 4: ***TOP**: Window-CER between hard-masking and soft-masing, **BOTTOM**: Window-CER using Down-sampling layer (DS) and Sinc filter*

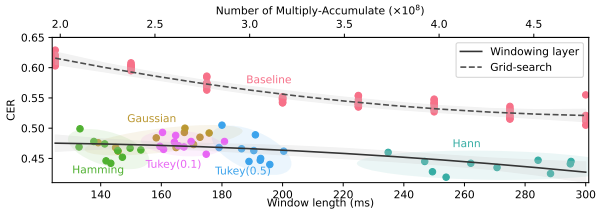

Figure 5: *Trade-off between CER and window length*

in Figure 6. $\lambda = 0.5$ and 1 provide the most energy-efficient performance, reducing the window length to 118 ms and sampling rate to 7.2 kHz while maintaining a CER of 0.49.

Table 1 shows comparisons between the proposed method, grid-search, and TPE. For raw audio, the proposed method is capable of reducing MAC by 73%, with comparable performance using Am-MobileNet. The energy used for data acquisition is also reduced by up to 57% for window-level speaker recognition. In the SincNet model, the proposed methods reduced the MAC in the model by 49% with a performance gain in window-level CER. Our proposed method is able to improve energy efficiency and CER in both models. However, CER and the energy utilized by SincNet are significantly higher than for AM-MobileNet, which is due to the DNN architecture itself.

To further improve power efficiency, the proposed method was evaluated on MFCC features using AM-MobileNet. The inference time, including MFCC extraction, was reduced from 12.7 ms to 8.2 ms with competitive CERs. Although speaker CERs using MFCC are higher than for raw audio, energy and time used at inference are reduced by half.

In most experimental setups, speaker CERs using our proposed method are lower than the baselines. The improved CERs may be due to the mechanism of optimizing the windowing layer, which allows other parameters in the model to learn on various receptive fields over the training epochs. This conjecture is evidenced by the result of the fixed value, which trains the model using fixed m and s values as in the proposed method, which has inferior results.

4.2. TBI detection

As reported in Table 1, the best TBI detection BA is obtained using 3.14s of speech sampled at 12.4 kHz, improving the grid-search result by 3.9%. The training time used to tune the model is significantly lower than TPE and is competitive with grid-search that trained DNN in parallel. Energy consumption at inference is expected to reduce by 26% compared to baseline. Similar to speaker recognition results, windowing and $\mathcal{D}_s$ layers allow the DNN to learn from different lengths and sampling rates of speech, which provides a better detection BA.

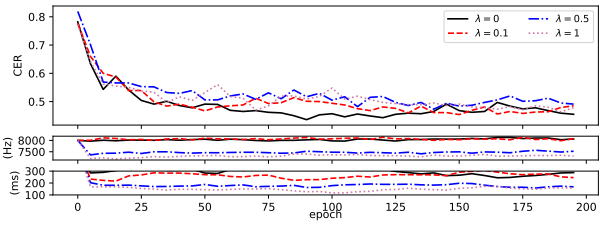

Figure 6: *Effect of energy-efficient penalty*

Table 1: *Speaker recognition and TBI detection results*

	Speaker classification	**CER** ($\%_{\pm std}$) Window	Sentence	**Energy-efficient metrics** m (ms)	s (Hz)	MAC	Inference Time (ms)	Training Time
SincNet (Raw audio)	$\mathcal{W}_{HM}$	**48.6**$_{.4}$	1.02$_{.13}$	**118**	8k	0.58	118	0.96
	$\mathcal{W}_{HM} + \mathcal{D}_s$	**49.1**$_{.3}$	1.08$_{.11}$	120	7.2k	**0.51**	**110**	1.05
	Grid-search	53.8$_{.1}$	**0.94**$_{.09}$	200	8k	1	184	1
	TPE	52.8$_{.7}$	1.29$_{.13}$	272	7.6k	0.87	135	22
	Fixed values	56.0$_{.1}$	1.24$_{.17}$	120	7.2k	0.51	118	0.64
Am-MobileNet (Raw audio)	$\mathcal{W}_G$	**21.9**$_{.1}$	0.36$_{.13}$	106	8k	0.46	16.7	1.13
	$\mathcal{W}_G + \mathcal{D}_s$	22.8$_{.2}$	**0.32**$_{.18}$	**99**	**5.1k**	**0.27**	**16.1**	1.02
	Grid-search	**21.4**$_{.1}$	0.38$_{.10}$	230	8k	1	23.4	1
	TPE	22.6$_{.2}$	0.39$_{.13}$	217	7.5k	0.88	23.5	17
	Fixed values	24.9$_{.2}$	0.38$_{.08}$	99	5.1k	0.46	16.1	0.49
Am-MobileNet (MFCC)	$\mathcal{W}_G$	**68.4**$_{.1}$	5.1$_{.08}$	**99**	8k	**0.76**	**8.2**	1.13
	$\mathcal{W}_{HM} + \mathcal{D}_s$	70.5$_{.2}$	6.3$_{.14}$	114	**7.1k**	**0.76**	8.8	1.19
	Grid-search	70.0$_{.1}$	**4.4**$_{.07}$	220	8k	1	12.7	1
	TPE	70.8$_{.2}$	4.9$_{.18}$	247	8k	1.0	12.7	14
	Fixed values	70.7$_{.1}$	8.9$_{0.19}$	99	8k	0.76	8.2	1.13
	TBI detection	**BA** ($\%_{\pm std}$)						
cGRU (Raw audio)	$\mathcal{W}_{HM}$	86.53$_{1.3}$		**2.89s**	8k	0.79	**3.7**s	1.05
	$\mathcal{W}_{HM} + \mathcal{D}_s$	**87.12**$_{1.4}$		3.14s	**6.2k**	**0.74**	**3.7**s	1.02
	Grid-search	83.82$_{1.4}$		4s	8k	1	4.6s	1
	TPE	81.90$_{1.9}$		3.94s	8k	1	4.6s	18
	Fixed values	82.62$_{1.1}$		3.14s	6.2k	0.74	3.7s	0.78

MAC and training time are reported as a ratio to Grid-search.

5. Conclusion

DNN-based speech processing has the potential for impact but currently has high energy consumption, limiting the mobile deployment of state-of-the-art methods. This study proposed learning an optimal speech length and sampling rate using a masking function during DNN back-propagation, which reduces the energy consumption of speech acquisition and DNN inference. Our evaluation demonstrates that learning speech format using an end-to-end model outperforms tuning window length and sampling rate as hyperparameters. As estimated using the MAC metric, the power consumption used for inference is reduced by up to 73% and 26% for the speaker recognition and TBI detection tasks, respectively, while maintaining high accuracy. Beyond the speaker recognition and TBI detection tasks, the proposed method is also broadly applicable to other speech tasks.

Our proposed method has the limitation of requiring subsequent DNN layers to operate on a tensor with a dynamic temporal dimension. The fully-connected DNN layer was modified to have a flexible input size, which may create inconsistent losses across training epochs. In the future, we plan to investigate additional speech processing parameters, such as hop size, to further reduce energy consumption of mobile speech processing.

6. References

[1] A. Ditthapron, A. C. Lammert, and E. O. Agu, "Continuous tbi monitoring from spontaneous speech using parametrized sinc filters and a cascading gru," J. Biomedical Health Inf., 2022.

[2] J. A. C. Nunes, D. Macêdo, and C. Zanchettin, "Am-mobilenet1d: A portable model for speaker recognition," in Proc. IJCNN. IEEE, 2020, pp. 1–8.

[3] J.-C. Wang, L.-X. Lian, Y.-Y. Lin, and J.-H. Zhao, "Vlsi design for svm-based speaker verification system," IEEE Trans. VLSI Systems, vol. 23, no. 7, pp. 1355–1359, 2014.

[4] H. Lu, A. Bernheim Brush, B. Priyantha, A. K. Karlson, and J. Liu, "Speakersense: Energy efficient unobtrusive speaker id. on mobile phones," in Int'l Conf. Pervasive Comp. Springer, 2011, pp. 188–205.

[5] B. Bergsma, M. Yang, and M. Cernak, "PEAF: Learnable Power Efficient Analog Acoustic Features for Audio Recognition," in Proc. Interspeech 2022, 2022, pp. 381–385.

[6] M. Ravanelli and Y. Bengio, "Speaker recognition from raw waveform with sincnet," in IEEE Spoken Language Tech. Workshop (SLT), 2018, pp. 1021–1028.

[7] K. Chlasta, K. Wołk, and I. Krejtz, "Automated speech-based screening of depression using deep convolutional neural networks," Procedia Computer Science, vol. 164, pp. 618–628, 2019.

[8] D. Oletic and V. Bilas, "System-level power consumption analysis of the wearable asthmatic wheeze quantification," Journal of Sensors, 2018.

[9] A. H. Victoria and G. Maragatham, "Automatic tuning of hyperparameters using bayesian optimization," Evolving Systems, vol. 12, no. 1, pp. 217–223, 2021.

[10] P. Bloomfield, Fourier analysis of time series: an introduction. John Wiley & Sons, 2004.

[11] D. W. Romero, R.-J. Bruintjes, J. M. Tomczak, E. J. Bekkers, M. Hoogendoorn, and J. van Gemert, "Flexconv: Continuous kernel convolutions with differentiable kernel sizes," in ICLR, 2022.

[12] R. Riad, O. Teboul, D. Grangier, and N. Zeghidour, "Learning strides in convolutional neural networks," in ICLR, 2022.

[13] Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," arXiv preprint arXiv:1308.3432, 2013.

[14] H. Bouraoui, C. Jerad, A. Chattopadhyay, and N. B. Hadj-Alouane, "Hardware architectures for embedded speaker recognition applications: a survey," ACM Transactions on Embedded Computing Systems (TECS), vol. 16, no. 3, pp. 1–28, 2017.

[15] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1," NASA STI/Recon technical report n, vol. 93, p. 27403, 1993.

[16] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in Proc. IEEE CVPR, 2018, pp. 4510–4520.

[17] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in Proceedings of the 14th python in science conference, vol. 8, 2015, pp. 18–25.

[18] C. A. Coelho, K. M. Youse, and K. N. Le, "Conversational discourse in closed-head-injured and non-brain-injured adults," Aphasiology, vol. 16, no. 4-6, pp. 659–672, 2002.

[19] X. He, K. Zhao, and X. Chu, "Automl: A survey of the state-of-the-art," Knowledge-Based Systems, vol. 212, p. 106622, 2021.

[20] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," NeurIPS, vol. 24, 2011.

[21] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," Proceedings of the IEEE, vol. 105, no. 12, pp. 2295–2329, 2017.