Exploring Filterbank Learning for Keyword Spotting

Iván López-Espejo[1], Zheng-Hua Tan[1], Jesper Jensen[1,2]
[1]*Department of Electronic Systems, Aalborg University, Denmark*
[2]*Oticon A/S, Denmark*
{ivl,zt,jje}@es.aau.dk, jesj@oticon.com

***Abstract*—Despite their great performance over the years, handcrafted speech features are not necessarily optimal for any particular speech application. Consequently, with greater or lesser success, optimal filterbank learning has been studied for different speech processing tasks. In this paper, we fill in a gap by exploring filterbank learning for keyword spotting (KWS). Two approaches are examined: filterbank matrix learning in the power spectral domain and parameter learning of a psychoacoustically-motivated gammachirp filterbank. Filterbank parameters are optimized jointly with a modern deep residual neural network-based KWS back-end. Our experimental results reveal that, in general, there are no statistically significant differences, in terms of KWS accuracy, between using a learned filterbank and handcrafted speech features. Thus, while we conclude that the latter are still a wise choice when using modern KWS back-ends, we also hypothesize that this could be a symptom of information redundancy, which opens up new research possibilities in the field of small-footprint KWS.**

***Index Terms*—Filterbank learning, keyword spotting, end-to-end, gammachirp filterbank, gammatone filterbank.**

I. Introduction

Handcrafted speech features such as Mel-frequency cepstral coefficients (MFCCs) and log-Mel features are well-established for many speech applications [1]. Those mimic human perception by roughly simulating aspects of the human auditory system and have shown good performance over the years. However, it is evident that these features are not necessarily optimal for any particular speech processing task and it is reasonable to believe that learned features could lead to better performance.

Thanks to the potentials of deep learning and the increasing availability of speech resources, a recent trend is the development of end-to-end deep learning systems where the feature extraction process is optimal according to the task and training criterion, e.g., [2], [3]. In particular, for applications like speaker verification anti-spoofing [4] and audio source separation and audio scene classification [5], optimal filterbank learning has shown improvements with respect to using a standard Mel filterbank.

Filterbank learning has also been explored for automatic speech recognition (ASR) purposes [6]–[8]. In [6], Sainath *et al.* train a raw time convolutional layer (i.e., filterbank), initialized with a gammatone filterbank, jointly with a convolutional, long short-term memory deep neural network (DNN) acoustic model. The front-end learned in [6], however, is not able to beat the performance of standard log-Mel features in terms of word error rate (WER). While the approach followed in [8] is very similar to that of [6], in [7], Seki *et al.* consider a pseudo-filterbank layer comprised of Gaussian-shaped filters operating in the power spectral domain. The gains, center frequencies and bandwidths of the pseudo-filterbank layer are trained jointly along with the back-end (i.e., DNN) for ASR. The improvements reported in [7], [8] are relatively modest and, moreover, it is unclear whether they are statistically significant, as the authors do not provide confidence intervals along with their WER results.

In this work, we explore filterbank learning for keyword spotting (KWS). To the best of our knowledge, [9] is the only (very recently) reported attempt that integrates filterbank learning in KWS. In [9], a convolutional neural network (CNN), which is trained to perform keyword prediction from the raw speech waveform, integrates parameterized sinc-convolutions acting as a filterbank. Such a front-end, in which only the cut-off frequencies of the filters are trainable along with the back-end parameters, was already proposed in [10]. Unfortunately, the authors of [9] do not carry out a comparison between using parameterized sinc-convolutions and traditional (i.e., handcrafted) speech features, so the possible advantages of employing a learned filterbank in terms of KWS performance remain unclear.

In this paper, we fill this gap by exploring two different filterbank learning approaches for KWS and comparing them with the use of traditional speech features. First, learning the weights of a filterbank matrix in the power spectral domain is examined. Secondly, we study the utilization of a psychoacoustically-motivated filterbank like the gammachirp [11] (which is an extension of the popular gammatone), where different parameters such as the gains, center frequencies and bandwidths of the filterbank are trainable. For both approaches, the learnable filterbank parameters are optimized by backpropagation jointly with a state-of-the-art KWS back-end consisting of a deep residual neural network [12].

From our experimental results, our main observation is that, in general, there are no statistically significant differences between the use of a learned filterbank and handcrafted speech features in terms of KWS accuracy, so we state that traditional speech features are still a good choice when employing modern KWS back-ends. Similarly, Robertson *et al.* [13] recently reported no statistically significant improvements to phone error rate when using either Gabor- or gammatone-based features instead of standard log-Mel features with a modern end-to-end CNN phone recognizer. In [13], they point

This work was supported, in part, by the Demant Foundation.

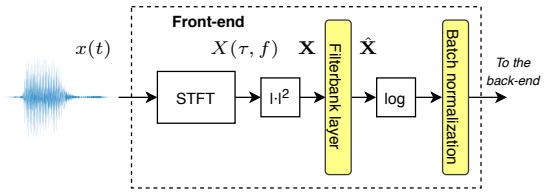

Fig. 1. Diagram of learnable filterbank matrix scheme.

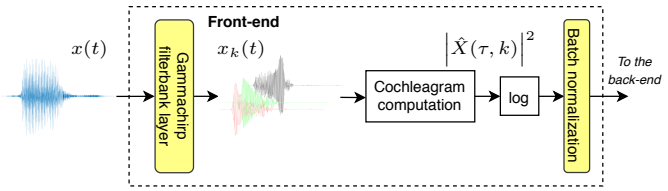

Fig. 2. Diagram of learnable gammachirp filterbank scheme.

out the difficulty comparing their work with previous work on learned filterbanks where single error rates are presented instead of statistical analyses of the results over repeated trials. The question is therefore whether those single error rates are meaningful or can be explained by a lucky setting of parameters.

The rest of this paper is organized as follows. In Section II, two different approaches for filterbank learning in the context of KWS are presented. The experimental framework is described in Section III. Then, our experimental results are shown and discussed in Section IV. Finally, Section V concludes this work.

II. Filterbank Learning for Keyword Spotting

In this section, we present two different filterbank learning approaches for KWS. Bear in mind that, for both approaches, the trainable filterbank parameters are optimized by backpropagation jointly with the deep residual neural network-based KWS back-end of [12] (architecture `res15`).

A. Filterbank Matrix Learning

Figure 1 depicts a diagram of our learnable filterbank matrix scheme. Notice that the front-end diagram is very similar to a log-Mel feature extraction front-end except that the Mel filterbank is replaced by a trainable filterbank.

Let $x(t)$ be a speech signal (possibly containing a keyword) and $X(\tau, f)$ its corresponding short-time Fourier transform (STFT), where $\tau = 1, ..., T$ and $f = 1, ..., F$ denote the time frame and linear frequency bin indices, respectively. In addition, T and F refer to the total number of time frames and linear frequency bins, respectively, of the signal. Let

$$\mathbf{X} = \begin{bmatrix} |X(1,1)|^2 & \dots & |X(1,F)|^2 \\ \vdots & \ddots & \vdots \\ |X(T,1)|^2 & \dots & |X(T,F)|^2 \end{bmatrix} \quad (1)$$

be a $T \times F$ matrix comprised of the squared magnitude of $X(\tau, f)$, then, the filterbank layer applies the following transform to $\mathbf{X}$:

$$\hat{\mathbf{X}} = \mathbf{X} \cdot h(\mathbf{W}), \quad (2)$$

where $\mathbf{W}$ is the learnable $F \times K$ filterbank matrix, K is the total number of filterbank channels and $h(\cdot)$ is an element-wise applied non-linearity to ensure the positivity of the filterbank weights (as similarly considered in, e.g., [4], [5]). In this work, $h(\cdot) = \max(\cdot, 0)$ is chosen to be the rectified linear unit (ReLU) function. Then, the result of the logarithmic compression $\log\left(\max\left(\hat{\mathbf{X}}, \eta\right)\right)$, where $\log(\cdot)$ and $\max(\cdot)$ are element-wise applied and $\eta = e^{-50}$ is a threshold to avoid numerical issues, is fed to a batch normalization layer the goal of which is to perform feature mean and variance normalization for robustness purposes. Finally, the output from the batch normalization layer is used by the back-end for keyword prediction.

B. Gammachirp Filterbank Learning

In this subsection, we consider a psychoacoustically-motivated gammachirp filterbank [11] with learnable parameters. This dynamic auditory filterbank consists of a gammatone filterbank with an additional frequency-modulation term, the so-called chirp term, that yields an asymmetric amplitude spectrum. The chirp term is coherent with physiological observations on frequency-modulations in mechanical responses of the basilar membrane [11].

The impulse responses of the gammachirp filterbank can be defined as [11]

$$g_c(t,k) = a_k t^{n-1} e^{-2\pi b \mathrm{ERB}(f_k) t} \times \cos\left(2\pi f_k t + c\log(t) + \phi\right), \quad (3)$$

where $\{a_k;\ k = 1, ..., K\}$ are filter gains, n and b define the envelope of the gamma function, c is the chirp term[1], ϕ is the initial phase (which is neglected in this work) and $\mathrm{ERB}(f_k)$ is the equivalent rectangular bandwidth of the k-th filter with center frequency f_k. At moderate stimulus levels [14],

$$\mathrm{ERB}(f_k) = 24.7 + 0.108 f_k \quad \text{[Hz]}. \quad (4)$$

A diagram of our learnable gammachirp filterbank scheme is outlined in Figure 2. The gammachirp filterbank layer implements the linear convolution operation $x_k(t) = x(t) * g_c(t,k)$ $(k = 1, ..., K)$, where a_k, n, b, c, f_k and the ERBs are trainable parameters. To preserve the physical meaning of these parameters, the ReLU function is applied to a_k, b, f_k and the ERBs, whereas n is constrained to be $\max(n, 1)$. Then, the cochleagram computation module segments every signal $x_k(t)$ into T overlapping frames of M samples each, $x_{\tau,k}(m)$ $(m = 1, ..., M)$, and estimates the cochleagram $\left|\hat{X}(\tau, k)\right|^2$ by means of Parseval's theorem as

$$\left|\hat{X}(\tau, k)\right|^2 = M \sum_{m=1}^{M} x_{\tau,k}^2(m). \quad (5)$$

Finally, logarithmic compression and batch normalization are applied to the cochleagram as discussed in Subsection II-A.

[1]Note that if $c = 0$, (3) becomes the gammatone filterbank.

III. Experimental Framework

We use the Google Speech Commands Dataset (GSCD) [15] for KWS experiments. This database consists of 105,829 one-second long speech files with a sampling rate of $f_s = 16$ kHz. Each speech file comprises one word among 35 possible candidate words. The GSCD is split into training (∼80% of the data), validation (∼10%) and test (∼10%) sets in such a manner that speakers do not overlap across sets. The deep residual neural network-based KWS back-end of [12] (architecture `res15`) is trained to spot the 10 keywords "yes", "no", "up", "down", "left", "right", "on", "off", "stop" and "go". Utterances with the remaining 25 words of the GSCD (i.e., non-keywords) are used to define the filler class, so the KWS back-end has to solve an 11-class classification problem. All the word classes are approximately balanced in the different sets.

The length of the analysis window and the hop size are, respectively, $M = 480$ and 160 samples (corresponding to 30 ms and 10 ms at $f_s = 16$ kHz). Therefore, every one-second long utterance is comprised of $T = 98$ time frames. Furthermore, $F = (M/2) + 1 = 241$ and, as is common [6]–[8], $K = 40$ is the number of filterbank channels.

The filterbank learning schemes presented in Section II and the KWS back-end are coded by means of Keras [16]. The back- and front-end are trained by using categorical cross-entropy as the loss function, and Adam [17] with default parameters as the optimizer (i.e., the learning rate is 0.001, $\beta_1 = 0.9$ and $\beta_2 = 0.999$). Similarly to [12], training runs for 26 epochs by default, which is found to be sufficient to guarantee convergence. The size of the minibatch is set to 64 training samples. During training, data augmentation is applied by carefully following the procedure described in [18].

As a KWS performance metric, we employ *accuracy*, which is defined as the ratio of the number of correct predictions over the total number of them. To draw meaningful conclusions, accuracy results are provided along with 95% confidence intervals calculated from outputs of 10 different back-end realizations trained with different random parameter initialization.

IV. Results and Discussion

A. Filterbank Matrix Learning

We first evaluate our learnable filterbank matrix scheme of Figure 1 by jointly and/or alternately training the back- and front-end for a number of epochs. The filterbank matrix of the filterbank layer, $\mathbf{W}$, is initialized by a Mel filterbank. It is worth to note that preliminary experiments explored the initialization of $\mathbf{W}$ by a linear-frequency spaced, triangular-shaped filterbank and no statistically significant differences were observed with respect to the Mel-based initialization.

Table I reports our KWS accuracy results from the learnable filterbank matrix scheme by following the naming convention FxBy_z, where x ∈ {t, f} indicates whether the front-end is trained, t, or not (i.e., fixed), f, y ∈ {t, f} indicates the same, but for the back-end, and z is the number of training epochs. Thus, we consider FfBt_26 a baseline, since it corresponds to the use of standard log-Mel features. As can be seen from Table I, jointly training the back- and front-end (i.e., the filterbank) from scratch, FtBt_26, does not yield a statistically significant improvement with respect to using standard log-Mel features. Therefore, we assess whether fine-tuning only the filterbank from our well-trained log-Mel baseline by 10 additional epochs, FfBt_26 + FtBf_10, provides some performance benefits. According to the results, this choice does not yield a statistically significant improvement, either (95.73% ± 0.38 *vs.* 95.64% ± 0.33 accuracy). This may be explained by the fact that the back-end is already optimized to work with a Mel filterbank, so substantially altering such a filterbank might even lead to worse performance. This hypothesis is supported by Figure 3, which plots the Mel filterbank and learned filterbanks from our learnable filterbank matrix scheme. In this figure, we can see at a glance the relatively higher similarity between the learned filterbank for FfBt_26 + FtBf_10 and the Mel filterbank. To no avail, we relax this constraint while still seizing the apparent virtues of the Mel filterbank by training only the back-end from scratch and, prior to convergence, jointly training the back- and front-end, FfBt_13 + FtBt_13.

TABLE I
KEYWORD SPOTTING ACCURACY RESULTS WITH 95% CONFIDENCE INTERVALS, IN PERCENTAGES, FROM OUR LEARNABLE FILTERBANK MATRIX SCHEME.

Test	Accuracy (%)
FfBt_26 (log-Mel)	95.64 ± 0.33
FtBt_26	95.73 ± 0.24
FfBt_26 + FtBf_10	95.73 ± 0.38
FfBt_13 + FtBt_13	95.30 ± 0.82

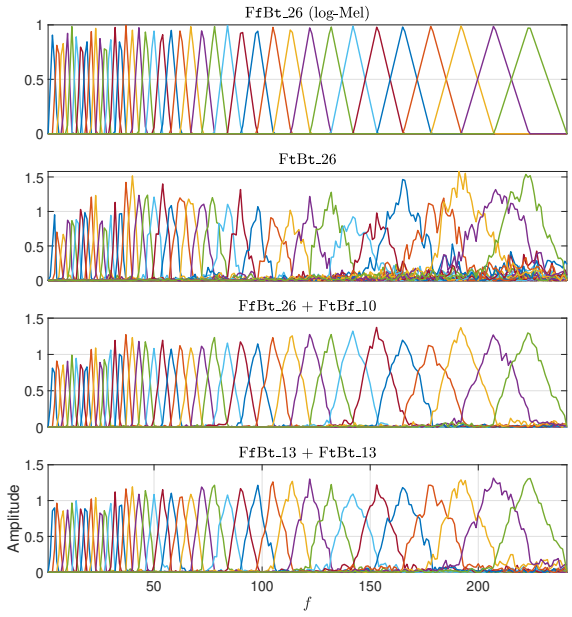

Fig. 3. Mel filterbank (top) and average (across the 10 experiment repetitions) learned filterbanks from our learnable filterbank matrix scheme.

TABLE II
KEYWORD SPOTTING ACCURACY RESULTS, IN PERCENTAGES, AND LEARNED n, b AND c VALUES FROM OUR LEARNABLE GAMMACHIRP FILTERBANK SCHEME. RESULTS ARE PROVIDED ALONG WITH 95% CONFIDENCE INTERVALS.

Test	Accuracy (%)	n	b	c
GT[f]_Ic-Mel	95.47 ± 0.36	4	1.019	0
GC[f]_Ic-Mel	95.45 ± 0.58	4	1.019	-1
GC[t]_Ic-Mel	95.12 ± 0.42	4.69 ± 0.07	0.976 ± 0.015	-0.84 ± 0.05
GC[t]_Ic-Linear	95.19 ± 0.52	4.44 ± 0.05	0.866 ± 0.019	-0.88 ± 0.02
GC[t]_Ir-Mel	94.68 ± 0.52	4.90 ± 0.51	0.976 ± 0.115	-0.97 ± 0.32
GC[t]_Ir-Linear	94.93 ± 0.45	4.65 ± 0.41	0.861 ± 0.075	-0.98 ± 0.38

TABLE III
KEYWORD SPOTTING ACCURACY RESULTS WITH 95% CONFIDENCE INTERVALS, IN PERCENTAGES, FROM FUSING LOG-MEL AND LEARNABLE GAMMACHIRP FEATURES AND REFERENCE TESTS.

Test	Accuracy (%)
FfBt_26 (log-Mel)	95.64 ± 0.33
GC[t]_Ic-Linear	95.19 ± 0.52
Fusion	95.65 ± 0.43

B. Gammachirp Filterbank Learning

Table II shows our KWS accuracy results and learned n, b and c values from our learnable gammachirp filterbank scheme of Figure 2. In this case, we follow the naming convention GC[x]_Iy-z, where x $\in \{$t, f$\}$ indicates whether the front-end is trained, t, or not[2], f, y $\in \{$c, r$\}$ refers to the initialization type of n, b and c which can be either constant, c, or random, r, and z tells whether the center frequencies f_k and the ERBs from (4) are initialized by a Mel or a linear scale[3]. When y $\equiv$ c, the initialization of the gamma function and chirp parameters is $n = 4$, $b = 1.019$ and $c = -1$ [11]. Otherwise, these parameters are initialized by uniform random sampling according to $n \sim \mathcal{U}(3, 5)$, $b \sim \mathcal{U}(0.8, 1.2)$ and $c \sim \mathcal{U}(-2, 0)$. The impulse responses of (3) are normalized to be in the range $[-1,\ 1]$ and a_k is initialized to 1 $\forall k$. Apart from a gammachirp baseline, GC[f]_Ic-Mel, a gammatone baseline, GT[f]_Ic-Mel, is also tested by simply setting $c = 0$.

From Table II, we can see that there are no statistically significant differences among the different tests in terms of KWS accuracy. Furthermore, standard deviations of the learned n, b and c parameters are larger for random initialization than for the constant one. This seems to indicate a certain sensitivity to initial values as well as there are no clear optimal n, b and c for the KWS task in terms of accuracy performance. In accordance with Figure 4, which shows the learned filter gains, center frequencies and ERBs from our learnable gammachirp filterbank scheme, this consideration is equally valid for these parameters, since Mel scale-based initialization leads to rather different learned parameters than the linear scale-based one.

In [6], max-pooling is employed for cochleagram derivation instead of (5). In this equation, notice that $x_{\tau,k}(m)$ results from segmentation of $x_k(t)$ by using a rectangular window. The authors of [8] claim that using a Hann window and the Parseval's theorem for cochleagram computation is superior to using max-pooling in the context of ASR. We have also tried these two approaches and no statistically significant differences were observed with respect to the approach reported in this paper.

Moreover, in [6], the learned front-end is unable to beat log-Mel features in terms of WER. The authors of [6] hypothesize that this can be due to the use of a strong back-end (i.e., acoustic model), though they finally find that this is not a reason when testing on lighter back-ends. Similarly, we explored the utilization of different lighter back-end models (e.g., res8-narrow [12]) and we observed the same KWS accuracy trends as the ones from using the stronger res15.

Finally, it is important to highlight that, unsuccessfully, we also tried to directly learn the impulse response samples as in [6], [8].

C. Feature Fusion

Sainath *et al.* [6] achieve to beat log-Mel features only by fusing the learnable front-end features with them. They argue that this is because of the complementarity of the learned and Mel filterbanks. As before, it is unclear if the reported improvement is statistically significant.

Table III presents the KWS accuracy result from fusing log-Mel features and GC[t]_Ic-Linear, as the linear scale-based initialization may help provide useful complementary information. As we can see, the fusion result is virtually identical to that from employing log-Mel features only, so we might conclude that the learned gammachirp filterbank conveys no additional information for KWS. Other fusion combinations lead to the same conclusion.

D. Filter Removal

Bearing in mind all of these results, a question emerges: is the filterbank and, in general, the speech feature design actually a crucial part of modern KWS systems? To study this question, we conduct KWS experiments using log-Mel features where we systematically remove filters from the filterbank in order to limit the amount of information available for keyword classification. Filterbank channel removal is carried out around channel $k = 23$, the center frequency of which is $f_{k=23} \approx 2{,}000$ Hz, since the frequency band contributing the most to human intelligibility is centered near 2,000 Hz [19]. Figure 5 plots KWS accuracy as a function of the range of removed filterbank channels. As can be seen from this figure, performance is negligibly affected even when removing the channels in the range $[20,\ 26]$ that spans, approximately, the frequency range from 1,626 Hz to 2,564 Hz. This result supports the hypothesis that KWS systems are fed with a

[2]In these experiments, the back-end is always trained.

[3]In [7], the trained center frequencies of the pseudo-filterbank layer hardly differ from their initialization. As the authors of [7] point out, this can be due to the big difference between the ranges of the center frequencies (i.e., $[0,\ 8{,}000]$ Hz) and other DNN weights. We tackle this issue by initializing the center frequencies normalized by $f_s/2$ and de-normalizing them prior to evaluating (3). A similar normalization procedure is carried out for the ERBs.

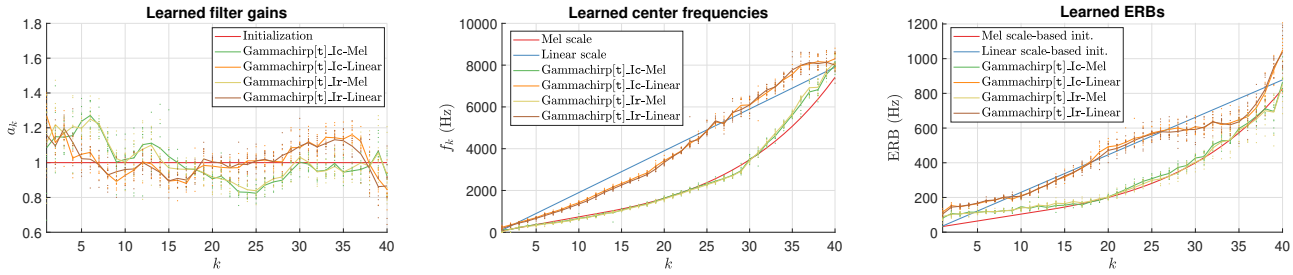

Fig. 4. Learned parameter values from our learnable gammachirp filterbank scheme as a function of the filterbank channel k. From left to right: filter gains a_k, center frequencies f_k and ERBs. Solid lines represent averages across the 10 experiment repetitions (points).

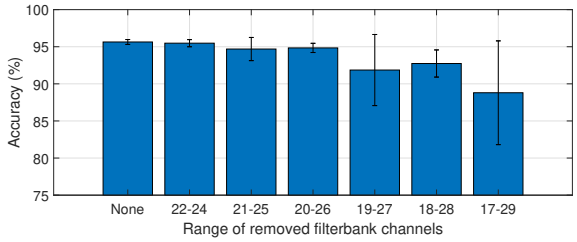

Fig. 5. Keyword spotting accuracy with 95% confidence intervals, in percentages, as a function of the range of removed filterbank channels for the test `FfBt_26` (log-Mel).

great amount of redundant information. Consequently, this gives clues on why the performance of learned filterbanks and traditional speech features is comparable.

V. Conclusion

In this paper, we have explored two different filterbank learning approaches for keyword spotting. Multiple experiments have shown that, in general, there are no statistically significant differences in terms of KWS accuracy between using a learned filterbank and handcrafted speech features, so we conclude that the latter are still a good choice when employing modern KWS back-ends. Furthermore, we have noticed that this could be a symptom of information redundancy, which opens up new research possibilities in the field of small-footprint (that is, low memory and computational complexity) KWS such as the design of much more compact speech features.

References

[1] B. Gold, N. Morgan, and D. Ellis, *Speech and Audio Signal Processing: Processing and Perception of Speech and Music (2nd Edition)*. Hoboken, New Jersey: John Wiley & Sons, 2011.

[2] J.-W. Jung, H.-S. Heo, I.-H. Yang, H.-J. Shim, and H.-J. Yun, "A complete end-to-end speaker verification system using deep neural networks: From raw signals to verification result," in *Proceedings of ICASSP 2018 – 43rd IEEE International Conference on Acoustics, Speech and Signal Processing, April 15-20, Calgary, Canada*, 2018, pp. 5349–5353.

[3] H. Muckenhirn, M. Magimai.-Doss, and S. Marcel, "Towards directly modeling raw speech signal for speaker verification using CNNs," in *Proceedings of ICASSP 2018 – 43rd IEEE International Conference on Acoustics, Speech and Signal Processing, April 15-20, Calgary, Canada*, 2018, pp. 4884–4888.

[4] H. Yu, Z.-H. Tan, Y. Zhang, Z. Ma, and J. Guo, "DNN filter bank cepstral coefficients for spoofing detection," *IEEE Access*, vol. 5, pp. 4779–4787, 2017.

[5] T. Zhang and J. Wu, "Discriminative frequency filter banks learning with neural networks," *EURASIP Journal on Audio, Speech, and Music*, pp. 1–16, 2019.

[6] T. N. Sainath, R. J. Weiss, A. Senior, K. W. Wilson, and O. Vinyals, "Learning the speech front-end with raw waveform CLDNNs," in *Proceedings of INTERSPEECH 2015 – 16th Annual Conference of the International Speech Communication Association, September 6-10, Dresden, Germany*, 2015, pp. 1–5.

[7] H. Seki, K. Yamamoto, and S. Nakagawa, "A deep neural network integrated with filterbank learning for speech recognition," in *Proceedings of ICASSP 2017 – 42nd IEEE International Conference on Acoustics, Speech and Signal Processing, March 5-9, New Orleans, USA*, 2017, pp. 5480–5484.

[8] N. Zeghidour, N. Usunier, G. Synnaeve, R. Collobert, and E. Dupoux, "End-to-end speech recognition from the raw waveform," in *Proceedings of INTERSPEECH 2018 – 19th Annual Conference of the International Speech Communication Association, September 2-6, Hyderabad, India*, 2018, pp. 781–785.

[9] S. Mittermaier, L. Kürzinger, B. Waschneck, and G. Rigoll, "Small-footprint keyword spotting on raw audio data with sinc-convolutions," *arXiv:1911.02086v1*, 2019.

[10] M. Ravanelli and Y. Bengio, "Interpretable convolutional filters with SincNet," *arXiv:1811.09725v2*, 2019.

[11] T. Irino and M. Unoki, "An analysis/synthesis auditory filterbank based on an IIR implementation of the gammachirp," *Journal of the Acoustical Society of Japan*, vol. 20, pp. 397–406, 1999.

[12] R. Tang and J. Lin, "Deep residual learning for small-footprint keyword spotting," in *Proceedings of ICASSP 2018 – 43rd IEEE International Conference on Acoustics, Speech and Signal Processing, April 15-20, Calgary, Canada*, 2018, pp. 5484–5488.

[13] S. Robertson, G. Penn, and Y. Wang, "Exploring spectro-temporal features in end-to-end convolutional neural networks," *arXiv:1901.00072v1*, 2019.

[14] B. Moore, R. Peters, and B. Glasberg, "Auditory filter shapes at low center frequencies," *Journal of the Acoustical Society of America*, vol. 88, pp. 132–140, 1990.

[15] P. Warden, "Speech Commands: A dataset for limited-vocabulary speech recognition," *arXiv:1804.03209v1*, 2018.

[16] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of ICLR 2015 – 3rd International Conference on Learning Representations, May 7-9, San Diego, USA*, 2015.

[18] I. López-Espejo, Z.-H. Tan, and J. Jensen, "Keyword spotting for hearing assistive devices robust to external speakers," in *Proceedings of INTERSPEECH 2019 – 20th Annual Conference of the International Speech Communication Association, September 15-19, Graz, Austria*, 2019, pp. 3223–3227.

[19] R. A. DePaolis, C. P. Janota, and T. Frank, "Frequency importance functions for words, sentences, and continuous discourse," *Journal of Speech, Language, and Hearing Research*, vol. 39, pp. 714–723, 1996.