Deep Spoken Keyword Spotting

An Overview

Espejo, Ivan Lopez; Tan, Zheng-Hua; Hansen, John; Jensen, Jesper

Published in:
IEEE Access

DOI (link to publication from Publisher):
10.1109/ACCESS.2021.3139508

Publication date:
2022

Document Version
Publisher's PDF, also known as Version of record

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 4, 2021, accepted December 28, 2021, date of publication December 30, 2021,
date of current version January 13, 2022.

Digital Object Identifier 10.1109/ACCESS.2021.3139508

Deep Spoken Keyword Spotting: An Overview

IVÁN LÓPEZ-ESPEJO[1], ZHENG-HUA TAN[1], (Senior Member, IEEE), JOHN H. L. HANSEN[2], (Fellow, IEEE), AND JESPER JENSEN[1,3]

[1]Department of Electronic Systems, Aalborg University, 9220 Aalborg, Denmark
[2]Erik Jonsson School of Engineering and Computer Science, The University of Texas at Dallas, Richardson, TX 75080, USA
[3]Oticon A/S, 2765 Smørum, Denmark

Corresponding author: Iván López-Espejo (ivl@es.aau.dk)

This work was supported in part by the Demant Foundation.

ABSTRACT Spoken keyword spotting (KWS) deals with the identification of keywords in audio streams and has become a fast-growing technology thanks to the paradigm shift introduced by deep learning a few years ago. This has allowed the rapid embedding of deep KWS in a myriad of small electronic devices with different purposes like the activation of voice assistants. Prospects suggest a sustained growth in terms of social use of this technology. Thus, it is not surprising that deep KWS has become a hot research topic among speech scientists, who constantly look for KWS performance improvement and computational complexity reduction. This context motivates this paper, in which we conduct a literature review into deep spoken KWS to assist practitioners and researchers who are interested in this technology. Specifically, this overview has a comprehensive nature by covering a thorough analysis of deep KWS systems (which includes speech features, acoustic modeling and posterior handling), robustness methods, applications, datasets, evaluation metrics, performance of deep KWS systems and audio-visual KWS. The analysis performed in this paper allows us to identify a number of directions for future research, including directions adopted from automatic speech recognition research and directions that are unique to the problem of spoken KWS.

INDEX TERMS Keyword spotting, deep learning, acoustic model, small footprint, robustness.

I. INTRODUCTION

Interacting with machines via voice is not science fiction anymore. Quite the opposite, speech technologies have become ubiquitous in nowadays society. The proliferation of voice assistants like Amazon's Alexa, Apple's Siri, Google's Assistant and Microsoft's Cortana is good proof of this [1]. A distinctive feature of voice assistants is that, in order to be used, they first have to be activated by means of a spoken wake-up word or keyword, thereby avoiding running far more computationally expensive automatic speech recognition (ASR) when it is not required [2]. More specifically, voice assistants deploy a technology called spoken keyword spotting —or simply keyword spotting—, which can be understood as a subproblem of ASR [3]. Particularly, keyword spotting (KWS) can be defined as the task of identifying keywords in audio streams comprising speech. And, apart from activating voice assistants, KWS has plenty of applications such as speech data mining, audio indexing, phone call routing, etc. [4].

Over the years, different techniques have been explored for KWS. One of the earliest approaches is based on the use of large-vocabulary continuous speech recognition (LVCSR) systems [5]–[7]. These systems are employed to decode the speech signal, and then, the keyword is searched in the generated lattices (i.e., in the representations of the different sequences of phonetic units that, given the speech signal, are likely enough). One of the advantages of this approach is the flexibility to deal with changing/non-predefined keywords [8]–[10] (although there is often a drop in performance when keywords are out of vocabulary [11]). The main disadvantage of LVCSR-based KWS systems might reside in the computational complexity dimension: these systems need to generate rich lattices, which requires high computational resources [9], [12] and also introduces latency [13]. While this should not be an issue for some applications like *offline* audio search [9], [14], LVCSR systems are not suitable for the lately-popular KWS applications[1] intended for small electronic devices (e.g., smartphones, smart speakers and

The associate editor coordinating the review of this manuscript and approving it for publication was Derek Abbott.

[1]By lately-popular KWS applications we mean activation of voice assistants, voice control, etc.

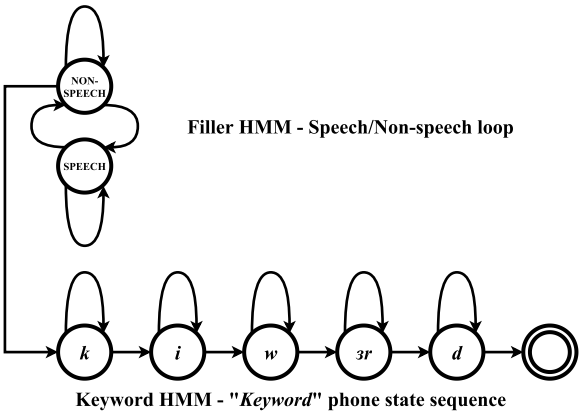

FIGURE 1. Scheme of a keyword/filler HMM-based KWS system [13] when the system keyword is *"keyword"*. While typically the keyword is modeled by a context-dependent triphone-based HMM, a monophone-based HMM is depicted instead for illustrative purposes. The filler HMM is often a speech/non-speech monophone loop.

wearables) characterized by notable memory, computation and power constraints [12], [15]–[17].

A still attractive and lighter alternative to LVCSR is the keyword/filler hidden Markov model (HMM) approach, which was proposed around three decades ago [18]–[20]. By this, a keyword HMM and a filler HMM are trained to model keyword and non-keyword audio segments, respectively, as illustrated by Figure 1. Originally, the acoustic features were modeled by means of Gaussian mixture models (GMMs) to produce the state emission likelihoods in keyword/filler HMM-based KWS [18]–[20]. Nowadays, similarly to the case of ASR, deep neural networks (DNNs) have replaced GMMs with this purpose [21]–[24] due to the consistent superior performance of the former. Viterbi decoding [25] is applied at runtime to find the best path in the decoding graph, and, whenever the likelihood ratio of the keyword model *versus* filler model is larger than a predefined threshold, the KWS system is triggered [13]. While this type of KWS systems is rather compact and good performing, it still needs Viterbi decoding, which, depending on the HMM topology, can be computationally demanding [12], [22].

The arrival of 2014 represented a milestone for KWS technology as a result of the publication of the first deep spoken KWS system [22]. In this paradigm (being new at the time), the sequence of word posterior probabilities yielded by a DNN is directly processed to determine the possible existence of keywords without the intervention of any HMM (see Figure 2). The deep KWS paradigm has recently attracted much attention [16], [26] due to a threefold reason:

1) It does not require a complicated sequence search algorithm (i.e., Viterbi decoding); instead, a significantly simpler posterior handling suffices;
2) The complexity of the DNN producing the posteriors (acoustic model) can be easily adjusted [9], [26] to fit the computational resource constraints;
3) It brings consistent significant improvements over the keyword/filler HMM approach in small-footprint (i.e., low memory and low computational complexity) scenarios in both clean and noisy conditions [17], [22].

This threefold reason makes it very appealing to deploy the deep KWS paradigm to a variety of consumer electronics with limited resources like earphones and headphones [27], smartphones, smart speakers and so on. Thus, much research on deep KWS has been conducted since 2014 until today, e.g., [15], [22], [26], [28]–[32]. And, what is more, we can expect that deep KWS will continue to be a hot topic in the future despite all the progress made.

In this paper, we present an overview of the deep spoken keyword spotting technology. We believe that this is a good time to look back and analyze the development trajectory of deep KWS to elucidate future challenges. It is worth noticing that only a small number of KWS overview articles is presently available in the literature [33]–[36]; at best, they shallowly encompass state-of-the-art deep KWS approaches, along with the most relevant datasets. Furthermore, while some relatively recent ASR overview articles covering acoustic modeling —which is a central part of KWS, see Figure 2— can also be found [37], [38], still (deep) KWS involves inherent issues, which need to be specifically addressed. *Some* of these inherent issues are related to posterior handling (see Figure 2), the class-imbalance problem [39], technology applications, datasets and evaluation metrics. To sum up, we can state that *1)* deep spoken KWS is currently a hot topic,[2] *2)* available KWS overview articles are outdated and/or they offer only a limited treatment of the latest progress, and *3)* deep KWS involves unique inherent issues compared to general-purpose ASR. Thus, this article aims at providing practitioners and researchers who are interested in the topic of keyword spotting with an up to date comprehensive overview of this technology.

The rest of this article is organized as follows: in Section II, the general approach to deep spoken KWS is introduced. Then, in Sections III, IV and V, respectively, the three main components constituting a modern KWS system are analyzed, i.e., speech feature extraction, acoustic modeling and posterior handling. In Section VI, we review current methods to strengthen the robustness of KWS systems against different sources of distortion. Applications of KWS are discussed in Section VII. Then, in Section VIII, we analyze the speech corpora currently employed for experimentally validating the latest KWS developments. The most important evaluation metrics for KWS are examined in Section IX. In Section X, a comparison among some of the latest deep KWS systems in terms of both KWS performance and computational complexity is presented. Section XI comprises a short review of the literature on audio-visual KWS. Finally, concluding remarks and comments about the future directions in the field are given in Section XII.

[2]A proof of this is the organization of events like Auto-KWS 2021 Challenge [40].

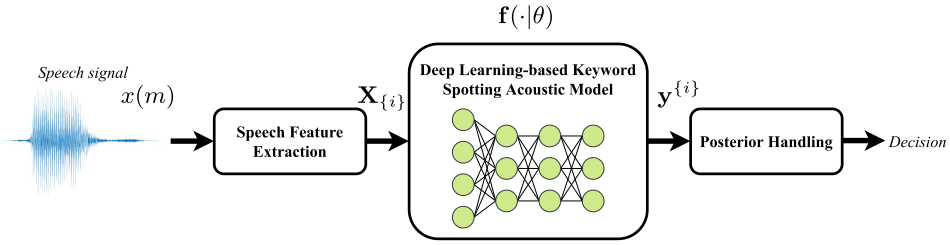

FIGURE 2. General pipeline of a modern deep spoken keyword spotting system: *1)* features are extracted from the speech signal, *2)* a DNN acoustic model uses these features to produce posteriors over the different keyword and filler (non-keyword) classes, and *3)* the temporal sequence of these posteriors is processed (Posterior Handling) to determine the possible existence of keywords.

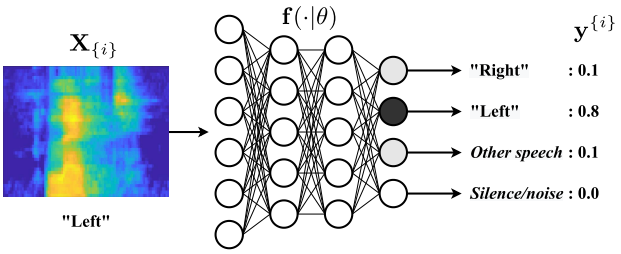

FIGURE 3. Illustrative example on how a DNN acoustic model performs. There are $N = 4$ different classes representing the keywords "right" and "left", *other speech* and *silence/noise*. The acoustic model receives a speech segment $\mathbf{X}_{\{i\}}$ (log-Mel spectrogram) comprising the keyword "left". The DNN produces a posterior distribution over the $N = 4$ different classes. Keyword "left" is given the highest posterior probability, 0.8.

II. DEEP SPOKEN KEYWORD SPOTTING APPROACH

Figure 2 depicts the general pipeline of a modern deep spoken keyword spotting system [15], [22], [28], [41]–[43], which is composed of three main blocks: *1)* the *speech feature extractor* converting the input signal to a compact speech representation, *2)* the *deep learning-based acoustic model* producing posteriors over the different keyword and filler (non-keyword) classes from the speech features (see the example of Figure 3), and *3)* the *posterior handler* processing the temporal sequence of posteriors to determine the possible existence of keywords in the input signal.

Let $x(m)$ be a finite acoustic time signal comprising speech. In the first place, the speech feature extractor computes an alternative representation of $x(m)$, namely, $\mathbf{X}$. It is desirable $\mathbf{X}$ to be *compact* (i.e., lower-dimensional, to limit the computational complexity of the task), *discriminative* in terms of the phonetic content and *robust* to acoustic variations [44]. Speech features $\mathbf{X}$ are traditionally represented by a two-dimensional matrix composed of a time sequence of K-dimensional feature vectors $\mathbf{x}_t$ $(t = 0, \ldots, T-1)$ as in

$$\mathbf{X} = (\mathbf{x}_0, \ldots, \mathbf{x}_t, \ldots, \mathbf{x}_{T-1}) \in \mathbb{R}^{K \times T}, \tag{1}$$

where T, the total number of feature vectors, depends on the length of the signal $x(m)$. Speech features $\mathbf{X}$ can be based on a diversity of representation types, such as, e.g., spectral [22], [28], [45], cepstral [16], [46] and time-domain ones [47]. Further details about the different types of speech features used for KWS are provided in Section III.

The DNN acoustic model receives $\mathbf{X}$ as input and outputs a sequence of posterior probabilities over the different keyword and non-keyword classes. Particularly, the acoustic model sequentially consumes time segments

$$\mathbf{X}_{\{i\}} = (\mathbf{x}_{is-P}, \ldots, \mathbf{x}_{is}, \ldots, \mathbf{x}_{is+F}) \tag{2}$$

of $\mathbf{X}$ until the whole feature sequence $\mathbf{X}$ is processed. In Eq. (2), $i = \lceil \frac{P}{s} \rceil, \ldots, \lfloor \frac{T-1-F}{s} \rfloor$ is an integer segment index and s represents the time frame shift. Moreover, P and F denote, respectively, the number of past and future frames (temporal context) in each segment $\mathbf{X}_{\{i\}} \in \mathbb{R}^{K \times (P+F+1)}$. While s is typically designed to have some degree of overlap between consecutive segments $\mathbf{X}_{\{i\}}$ and $\mathbf{X}_{\{i+1\}}$, many works consider acoustic models classifying non-overlapping segments that are sufficiently long (e.g., one second) to cover an entire keyword [16], [30], [48]–[53]. With regard to P and F, a number of approaches considers $F < P$ to reduce latency without significantly sacrificing performance [12], [22], [28], [41]. In addition, voice activity detection [54] is sometimes used to reduce power consumption by only inputting to the acoustic model segments $\mathbf{X}_{\{i\}}$ in which voice is present [11], [22], [55]–[57].

Then, let us suppose that the DNN acoustic model $\mathbf{f}(\cdot|\theta) : \mathbb{R}^{K \times (P+F+1)} \rightarrow I^N$ has N output nodes meaning N different classes, where θ and $I = [0, 1]$ denote the parameters of the acoustic model and the unit interval, respectively. Normally, the output nodes represent either words [12], [16], [22], [28], [30], [41], [43], [48]–[53], [57]–[59] or subword units like context-independent phonemes [31], [60]–[62], the latter especially in the context of sequence-to-sequence models [63]–[65] (see Subsection IV-C for further details). Let subscript n refer to the n-th element of a vector. For every input segment $\mathbf{X}_{\{i\}}$, the acoustic model yields

$$\mathbf{y}_n^{\{i\}} = \mathbf{f}_n\left(\mathbf{X}_{\{i\}} \middle| \theta\right), \quad n = 1, \ldots, N, \tag{3}$$

where $\mathbf{y}_n^{\{i\}} = P\left(C_n | \mathbf{X}_{\{i\}}, \theta\right)$ is the posterior of the n-th class C_n given the input feature segment $\mathbf{X}_{\{i\}}$. To ensure that $\sum_{n=1}^{N} \mathbf{y}_n^{\{i\}} = 1 \quad \forall i$, deep KWS systems commonly employ a fully-connected layer with softmax activation [66] as an output layer, e.g., [16], [43], [47], [52], [60], [67]–[72]. The parameters of the model, θ, are usually estimated by discriminatively training $\mathbf{f}(\cdot|\theta)$ by backpropagation from

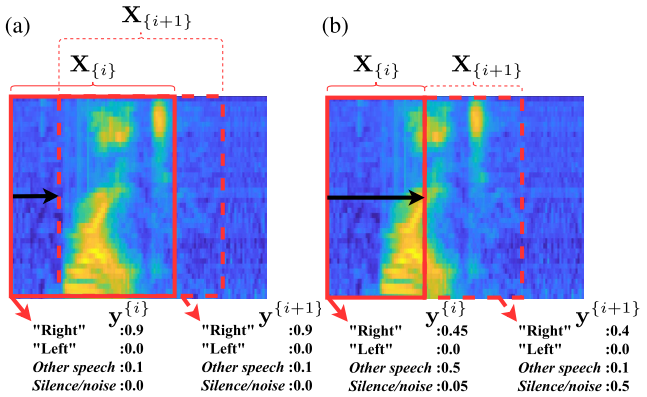

FIGURE 4. Example of the processing of two consecutive feature segments $\mathbf{X}_{\{i\}}$ and $\mathbf{X}_{\{i+1\}}$, from $\mathbf{X}$ comprising the keyword "right", by a DNN acoustic model: (a) when using an overlapping segmentation window, and (b) when using a smaller, non-overlapping one.

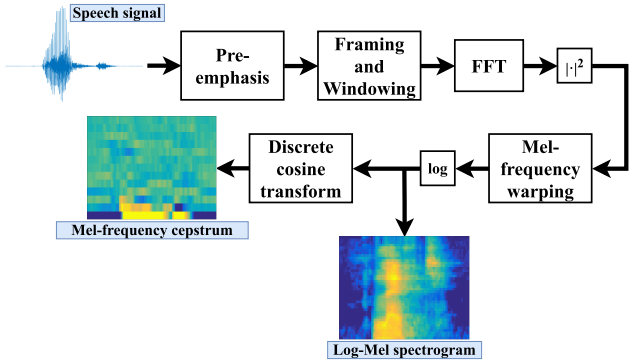

FIGURE 5. Classical pipeline for extracting log-Mel spectral and Mel-frequency cepstral speech features using the fast Fourier transform (FFT).

annotated speech data characterizing the different N classes. The most popular loss function that is employed to this end is cross-entropy loss [73], [74].

Figure 3 shows an example, illustrating the above paragraph, in which there are $N = 4$ different classes. Two of these classes represent the keywords "right" (C_1) and "left" (C_2). The other two classes are the filler classes *other speech* (C_3) and *silence/noise* (C_4). A segment $\mathbf{X}_{\{i\}}$ consisting of a log-Mel spectrogram comprising the keyword "left" is input to the DNN acoustic model. Then, this generates a posterior distribution $\mathbf{y}^{\{i\}}$ over the $N = 4$ classes. Keyword "left" is given the highest posterior probability, namely, $\mathbf{y}_2^{\{i\}} = P\left(C_2|\,\mathbf{X}_{\{i\}}, \theta\right) = 0.8$.

Most of the research that has been conducted on deep KWS has focused on its key part, which is the design of increasingly accurate and decreasingly computationally complex acoustic models $\mathbf{f}(\cdot|\theta)$ [32], [75].

Finally, KWS is not a static task but a dynamic one in which the KWS system has to continuously listen to the input signal $x(m)$ to yield the sequence of posteriors $\mathbf{y}^{\{i\}}$, $i = \lceil \frac{P}{s} \rceil, \ldots, \lfloor \frac{T-1-F}{s} \rfloor$, in order to detect keywords in real-time. In the example in Figure 3, a straightforward way to do this could just be choosing the class $\hat{C}^{\{i\}}$ with the highest posterior, that is,

$$\hat{C}^{\{i\}} = \underset{C_n}{\operatorname{argmax}}\, \mathbf{y}_n^{\{i\}} = \underset{C_n}{\operatorname{argmax}}\, P\left(C_n|\,\mathbf{X}_{\{i\}}, \theta\right). \tag{4}$$

Nevertheless, this approach is not robust, as discussed in what follows. Continuing with the illustration of Figure 3, Figure 4 exemplifies the processing by the acoustic model of two consecutive feature segments $\mathbf{X}_{\{i\}}$ and $\mathbf{X}_{\{i+1\}}$ from $\mathbf{X}$ comprising the keyword "right". Figure 4a shows the typical case of using an overlapping segmentation window. As we can see, following the approach of Eq. (4) might lead to detecting the same keyword realization twice, yielding a false alarm. In addition, Figure 4b depicts the case in which a non-overlapping segmentation window is employed. In this situation, the energy of the keyword realization leaks into two different segments in such a manner that neither the posterior $P\left(C_1|\,\mathbf{X}_{\{i\}}, \theta\right)$ nor $P\left(C_1|\,\mathbf{X}_{\{i+1\}}, \theta\right)$ is sufficiently strong for the keyword to be detected, thereby yielding a miss detection. Hence, a proper handling of the sequence of posteriors $\mathbf{y}^{\{i\}}$ ($i = \lceil \frac{P}{s} \rceil, \ldots, \lfloor \frac{T-1-F}{s} \rfloor$) is a very important component for effective keyword detection [2], [4], [15], [22], [29], [41]–[43], [45], [46], [56], [76]–[79]. Posterior handling is examined in Section V.

III. SPEECH FEATURE EXTRACTION

In the following subsections, we walk through the most relevant speech features revolving around deep KWS: Mel-scale-related features, recurrent neural network features, low-precision features, learnable filterbank features and other features.

A. MEL-SCALE-RELATED FEATURES

Speech features based on the perceptually-motivated Mel-scale filterbank [80], like the log-Mel spectral coefficients and Mel-frequency cepstral coefficients (MFCCs) [81], have been widely used over decades in the fields of ASR and, indeed, KWS. Despite the multiple attempts to learn optimal, alternative representations from the speech signal (see Subsection III-D for more details), Mel-scale-related features are still nowadays a solid, competitive and safe choice [82]. Figure 5 depicts the well-known classical pipeline for extracting log-Mel spectral and MFCC features. In deep KWS, both types of speech features are commonly normalized to have zero mean and unit standard deviation before being input to the acoustic model, thereby stabilizing and speeding up training as well as improving model generalization [83].

Mel-scale-related features are, by far, the most widely used speech features in deep KWS. For example, MFCCs with temporal context and, sometimes, their first- and second-order derivatives are used in [16], [30], [46], [51]–[53], [84]–[91]. As can be seen from Figure 5, MFCCs are obtained from the application of the discrete cosine transform to the log-Mel spectrogram. This transform produces approximately decorrelated features, which are

well-suited to, e.g., acoustic models based on GMMs that, for computational efficiency reasons, use diagonal covariance matrices. However, deep learning models are able to exploit spectro-temporal correlations, yielding the use of the log-Mel spectrogram instead of MFCCs equivalent or better ASR and KWS performance [92]. As a result, a good number of deep KWS works considers log-Mel or Mel filterbank speech features with temporal context, e.g., [8], [9], [15], [22], [26], [28], [29], [31], [43], [45], [48], [55], [58], [60], [62], [68], [71], [72], [78], [93]–[101]. In addition, [79] proposes instead the use of the first derivative of the log-Mel spectrogram to improve robustness against signal gain changes. The number of filterbank channels in the above works ranges from 20 to 128. In spite of this wide channel range, experience suggests that (deep) KWS performance is not significantly sensitive to the value of this parameter as long as the Mel-frequency resolution is not very poor [82]. This fact could promote the use of a lower number of filterbank channels in order to limit computational complexity.

B. RECURRENT NEURAL NETWORK FEATURES

Recurrent neural networks (RNNs) are helpful to summarize variable-length data sequences into fixed-length, compact feature vectors, also known as *embeddings*. Due to this fact, RNNs are very suitable for template matching problems like query-by-example (QbE) KWS, which involves keyword detection by determining the similarity between feature vectors (successively computed from the input audio stream) and keyword templates. In, e.g., [11], [56], [102]–[104], long short-term memory (LSTM) and gated recurrent unit (GRU) neural networks are employed to extract word embeddings. Generally, these are compared, by means of any distance function like cosine similarity [105] and particularly for QbE KWS, with keyword embeddings obtained during an enrollment phase.

While QbE KWS based on RNN feature extraction —which is different from the approach outlined in Section II and requires a careful treatment of its specificities— is out of the scope of this paper, we have considered it pertinent to allude to it for the following twofold reason. First, there is little difference between the general pipeline of Figure 2 and QbE KWS based on RNN feature extraction, since acoustic modeling is implicitly carried out by the RNN.[3] Second, QbE KWS based on RNN feature extraction is especially useful for personalized, open-vocabulary KWS, by which a user is allowed to define her/his own keywords by just recording a few keyword samples during an enrollment phase. Alternatively, in [103], a clever RNN mechanism to generate keyword templates from text instead of speech inputs is proposed. Notice that incorporating new keywords in the context of the deep spoken KWS approach introduced in Section II might require system re-training, which is not always feasible.

QbE KWS based on RNN feature extraction has shown to be more efficient and better performing than classical QbE KWS approaches based on LVCSR [106] and dynamic time warping (DTW) [107]. Therefore, the RNN feature approach is a good choice for on-device KWS applications providing keyword personalization.

C. LOW-PRECISION FEATURES

A way to diminish the energy consumption and memory footprint of deep KWS systems to be run on resource-constrained devices consists, e.g., of quantization —i.e., precision reduction— of the acoustic model parameters. Research like [108], [109] has demonstrated that it is possible to (closely) achieve the accuracy provided by full-precision acoustic models while drastically decreasing memory footprint by means of 4-bit quantization of model's weights.

The same philosophy can be applied to speech features. Emerging research [69] studies two kinds of low-precision speech representations: linearly-quantized log-Mel spectrogram and power variation over time, derived from log-Mel spectrogram, represented by only 2 bits. Experimental results show that using 8-bit log-Mel spectra yields same KWS accuracy as employing full-precision MFCCs. Furthermore, KWS performance degradation is insignificant when exploiting 2-bit precision speech features. As the authors of [69] state, this fact might indicate that much of the spectral information is superfluous when attempting to spot a set of keywords. In [82], we independently arrived at the same finding. In conclusion, there appears to be a large room for future work on the design of new extremely-light and compact (from a computational point of view) speech features for small-footprint KWS (see also the next subsection).

D. LEARNABLE FILTERBANK FEATURES

The development of end-to-end deep learning systems in which feature extraction is optimal in line with the task and training criterion is a recent trend (e.g., [110], [111]). This approach aspires to become an alternative to the use of well-established handcrafted features like log-Mel features and MFCCs, which are preferred for many speech-related tasks, including deep KWS (see Subsection III-A).

Optimal filterbank learning is part of such an end-to-end training strategy, and it has been explored for deep KWS in [70], [82]. In this context, filterbank parameters are tuned towards optimizing word posterior generation. Particularly, in [70], the acoustic model parameters are optimized jointly with the cut-off frequencies of a filterbank based on sinc-convolutions (SincConv) [112]. Similarly, in [82], we studied two filterbank learning approaches: one consisting of filterbank matrix learning in the power spectral domain and another based on parameter learning of a psychoacoustically-motivated gammachirp filterbank [113]. While the use of SincConv is not compared with using handcrafted speech features in [70], in [82], we found no

[3] Actually, in [11], [102], the LSTM networks used there are pure acoustic models, and the word embeddings correspond to the activations prior to the output softmax layer.

statistically significant KWS accuracy differences between employing a learned filterbank and log-Mel features. This finding is in line with research on filterbank learning for ASR, e.g., [114]–[116]. In [82], it is hypothesized that such a finding might be an indication of information redundancy.[4] As suggested in Subsection III-C, this should encourage research on extremely-light and compact speech features for small-footprint KWS. In conclusion, handcrafted speech features currently provide state-of-the-art KWS performance at the same time that optimal feature learning requires further research to become the preferred alternative.

E. OTHER SPEECH FEATURES

A small number of works has explored the use of alternative speech features with a relatively low computational impact. For example, [47] introduced the so-called multi-frame shifted time similarity (MFSTS). MFSTS are time-domain features consisting of a two-dimensional speech representation comprised of constrained-lag autocorrelation values. Despite their computational simplicity, which can make them attractive for low-power KWS applications, features like MFCCs provide much better KWS accuracy [47].

A more interesting approach is that examined by [117], [118], which fuses two different KWS paradigms: DTW and deep KWS. First, a DTW warping matrix measuring the similarity between an input speech utterance and the keyword template is calculated. From the deep KWS perspective, this matrix can be understood as speech features that are input to a deep learning binary (i.e., keyword/non-keyword) classifier playing the role of an ''acoustic model''. This hybrid approach brings the best of both worlds: *1)* the powerful modeling capabilities of deep KWS, and *2)* the flexibility of DTW KWS to deal with both open-vocabulary and language-independent scenarios. In spite of its potentials, further research on this methodology is needed, since, e.g., it is prone to overfitting [118].

IV. ACOUSTIC MODELING

This section is devoted to review the core of deep spoken KWS systems: the acoustic model. The natural trend is the design of increasingly accurate models while decreasing computational complexity. In an approximate chronological order, Subsections IV-A, IV-B and IV-C review advances in acoustic modeling based on fully-connected feedforward networks, convolutional networks, and recurrent and time-delay neural networks, respectively. Finally, Subsection IV-D is dedicated to how these acoustic models are trained.

A. FULLY-CONNECTED FEEDFORWARD NEURAL NETWORKS

Deep spoken KWS made its debut in 2014 [22] employing acoustic modeling based on the most widespread type of neural architecture at the time: the fully-connected feedforward neural network (FFNN). A simple stack of three fully-connected hidden layers with 128 neurons each and rectified linear unit (ReLU) activations, followed by a softmax output layer, greatly outperformed, with fewer parameters, a (at that time) state-of-the-art keyword/filler HMM system in both clean and noisy acoustic conditions. However, since the constant goal is the design of more accurate/robust and computationally lighter acoustic models, the use of fully-connected FFNNs was quickly relegated to a secondary level. Nowadays, state-of-the-art acoustic models use convolutional and recurrent neural networks (see Subsections IV-B and IV-C), since they can provide better performance with fewer parameters, e.g., [9], [28]. Even so, standard FFNN acoustic models and variants of them[5] are considered in recent literature for either comparison purposes or studying different aspects of KWS such as training loss functions, e.g., [9], [17], [42], [56].

Closely related and computationally cheaper alternatives to fully-connected FFNNs are single value decomposition filter (SVDF) [31], [71], [119] and spiking neural networks [41], [53], [120]. Proposed in [119] to approximate fully-connected layers by low-rank approximations, SVDF achieved to reduce by 75% the FFNN acoustic model size of the first deep KWS system [22] with no drop in performance. A similar idea was explored in [121], where a high degree of acoustic model compression is accomplished by means of low-rank weight matrices. The other side of the same coin is that modeling power can be enhanced by increasing the number of neurons while keeping the original number of multiplications fixed [121]. In this way, the performance of the first deep KWS system [22] was improved without substantially altering the computational resource usage of the algorithm. Higuchi *et al.* [59] have shown that an SVDF neural network is a special case of a stacked one-dimensional convolutional neural network (CNN), so the former can be easily implemented as the latter.

On the other hand, spiking neural networks (SNNs) are human brain-inspired neural networks that, in contrast to artificial neural networks (ANNs), process the information in an event-driven manner, which greatly alleviates the computational load when such information is sparse as in KWS [41], [53], [120]. To make them work, in the first place, real-valued input data like speech features have to be transformed to a sequence of spikes encoding real values in either its frequency (spike rate) or the relative time between spikes. Then, spikes propagate throughout the SNN to eventually fire the corresponding output neurons, which represent word classes in KWS [41]. SNNs can yield a similar KWS performance to that of equivalent ANNs while providing a computational cost reduction and energy saving above 80% [41] and of dozens of times [53], respectively. Apart from having been applied

[4] With a sufficiently powerful DNN acoustic model, the actual input feature representation is of less importance (as long as it represents the relevant information about the input signal).

[5] For example, in [32], it is evaluated an FFNN acoustic model integrating an intermediate pooling layer, which yields improved KWS accuracy in comparison with a standard FFNN using a similar number of parameters.

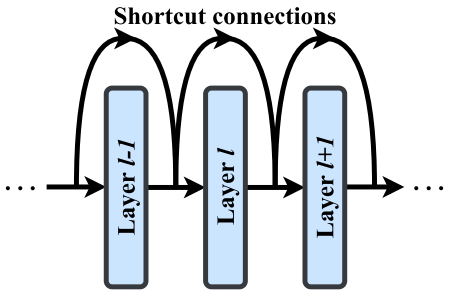

FIGURE 6. Example of shortcut connections linking non-consecutive layers in residual learning models.

to fully-connected FFNNs for KWS [41], [53], the SNN paradigm has also been recently applied to CNN acoustic modeling [53], which is reviewed in the next subsection.

B. CONVOLUTIONAL NEURAL NETWORKS

Moving from fully-connected FFNN to CNN acoustic modeling was a natural step taken back in 2015 [28]. Thanks to exploiting local speech time-frequency correlations, CNNs are able to outperform, with fewer parameters, fully-connected FFNNs for acoustic modeling in deep KWS [28], [32], [72], [86], [96], [117], [122]–[125]. One of the attractive features of CNNs is that the number of multiplications of the model can be easily limited to meet the computational constraints by adjusting different hyperparameters like, e.g., filter striding, and kernel and pooling sizes. Moreover, this may be done without necessarily sacrificing much performance [28].

Residual learning, proposed by He *et al.* [126] for image recognition, is widely considered to implement state-of-the-art acoustic models for deep KWS [30], [32], [50]–[52], [57], [67], [69], [78]. In short, residual learning models are constructed by introducing a series of shortcut connections linking non-consecutive layers (as exemplified by Figure 6), which helps to better train very deep CNN models. To the best of our knowledge, Tang and Lin [30] were the first authors exploring deep residual learning for deep KWS. They also integrated dilated convolutions increasing the network's receptive field in order to capture longer time-frequency patterns[6] without increasing the number of parameters, as also done by a number of subsequent deep KWS systems, e.g., [47], [51], [78]. In this way, Tang and Lin greatly outperformed, with less parameters, standard CNNs [28] in terms of KWS performance, establishing a new state-of-the-art back in 2018. Their powerful deep residual architecture so-called `res15` has been employed to carry out different KWS studies in areas like robustness for hearing assistive devices [128], [129], filterbank learning [82], and robustness to acoustic noise [130], among others.

Largely motivated by this success, later work further explored the use of deep residual learning. For example, [67] uses a variant of DenseNet [131], which can be interpreted as an extreme case of residual network comprising a hive of skip connections and requiring fewer parameters. The use of an acoustic model inspired by WaveNet [132], involving both skip connections and gated activation units, is evaluated in [78]. Choi *et al.* [50] proposed utilizing one-dimensional convolutions along the time axis (*temporal convolutions*) while treating the (MFCC) features as input channels within a deep residual learning framework (TC-ResNet). This approach could help to overcome the challenge of simultaneously capturing both high and low frequency features by means of not very deep networks —although we think that this can also be accomplished, to a great extent, by two-dimensional dilated convolutions increasing the network's receptive field—. The proposed temporal convolution yields a significant reduction of the computational burden with respect to a two-dimensional convolution with the same number of parameters. As a result, TC-ResNet matches Tang and Lin's [30] KWS performance while dramatically decreasing both latency and the amount of floating-point operations per second on a mobile device [50]. In [32], where an interesting deep KWS system comparison is presented, TC-ResNet, exhibiting one of the least latency and model sizes, is top-ranked in terms of KWS performance, outperforming competitive acoustic models based on standard CNNs, convolutional recurrent neural networks (CRNNs) [75], and RNNs with an attention mechanism [133] (see also the next subsection), among others. Furthermore, very recently, Zhou *et al.* [134] adopted a technique so-called AdderNet [135] to replace multiplications by additions in TC-ResNet, thereby drastically reducing its power consumption while maintaining a competitive accuracy.

Another appealing way to reduce the computation and size of standard CNNs is by depthwise separable convolutions [136]. They work by factorizing a standard convolution into a depthwise one and a pointwise (1×1) convolution combining the outputs from the depthwise one to generate new feature maps [136]. Depthwise separable CNNs (DS-CNNs) are a good choice to implement well-performing acoustic models in embedded systems [43], [45]. For example, the authors of [70] are able to reproduce the outstanding performance of TC-ResNet [50] using less parameters thanks to exploiting depthwise separable convolutions. Furthermore, the combination of depthwise separable convolutions with residual learning has been recently explored for deep KWS acoustic modeling [51], [52], [57], [100], generally outperforming all standard residual networks [30], plain DS-CNNs and TC-ResNet with less computational complexity.

Upon this review, we believe that a modern CNN-based acoustic model should ideally encompass the following three aspects:

1) A mechanism to exploit long time-frequency dependencies like, e.g., the use of temporal convolutions [50] or dilated convolutions.
2) Depthwise separable convolutions [136] to substantially reduce both the memory footprint and computation of the model without sacrificing the performance.

[6] In [49], the authors achieve this same effect by means of graph convolutional networks [127].

3) Residual connections [126] to fast and effectively train deeper models providing enhanced KWS performance.

C. RECURRENT AND TIME-DELAY NEURAL NETWORKS

Speech is a temporal sequence with strong time dependencies. Therefore, the utilization of RNNs for acoustic modeling —and also time-delay neural networks (TDNNs), which are shaped by a set of layers performing on different time scales— naturally arises. For example, LSTM neural networks [137], which overcome the exploding and vanishing gradient problems suffered by standard RNNs, are used for KWS acoustic modeling in, e.g., [4], [29], [76], [78], [84], clearly outperforming FFNNs [29]. When latency is not a strong constraint, bidirectional LSTMs (BiLSTMs) can be used instead to capture both causal and anticausal dependencies for improved KWS performance [76], [138]. Alternatively, bidirectional GRUs are explored in [32] for KWS acoustic modeling. When there is no need to model very long time dependencies, as it is the case in KWS, GRUs might be preferred over LSTMs since the former demand less memory and are faster to train while performing similarly or even better [93].

Besides, [58] studies a two-stage TDNN consisting of an LVCSR acoustic model followed by a keyword classifier. The authors of [58] also investigate the integration of frame skipping and caching to decrease computation, thereby outperforming classical CNN acoustic modeling [28] while halving the number of multiplications.

As we already suggested in Subsection IV-B, CNNs might have difficulties to model long time dependencies. To overcome this point, they can be combined with RNNs to build the so-called CRNNs. Thus, it may be stated that CRNNs bring the best of two worlds: first, convolutional layers model local spectro-temporal correlations of speech and, then, recurrent layers follow suit by modeling long-term time dependencies in the speech signal. Some works explore the use of CRNNs for acoustic modeling in deep spoken KWS using either unidirectional or bidirectional LSTMs or GRUs [32], [48], [76], [93], [109], [118]. Generally, the use of CRNNs allows us for outperforming standalone CNNs and RNNs [48].

1) CONNECTIONIST TEMPORAL CLASSIFICATION

As for the majority of acoustic models, the above-reviewed RNN acoustic models are typically trained to produce frame-level posterior probabilities. At training time, in case of employing, e.g., cross-entropy loss, frame-level annotated data are required, which may be cumbersome to get. In the context of RNN acoustic modeling, connectionist temporal classification (CTC) [63] is an attractive alternative letting the model unsupervisedly locate and align the phonetic unit labels at training time [4]. In other words, frame-level alignments of the target label sequences are not required for training.

Mathematically speaking, let $\mathbf{C} = (c_0, \ldots, c_{m-1})$ be the sequence of phonetic units or, e.g., characters corresponding to the sequence of feature vectors $\mathbf{X} = (\mathbf{x}_0, \ldots, \mathbf{x}_{T-1})$, where

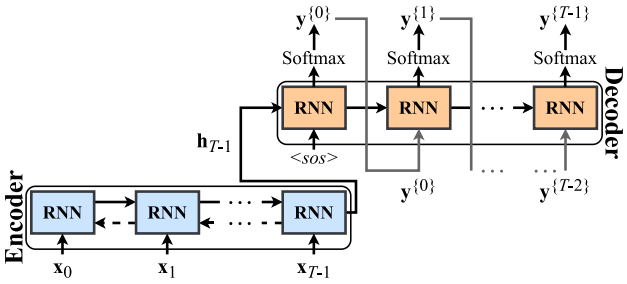

FIGURE 7. Example of sequence-to-sequence (Seq2Seq) model. Here, "*<sos>*" stands for "start of sequence". See the text for further details.

$m < T$ and the accurate alignment between $\mathbf{C}$ and $\mathbf{X}$ is unknown. CTC is an *alignment-free* algorithm whose goal is to maximize [63]

$$P(\mathbf{C}|\mathbf{X}) = \sum_{A \in \mathcal{A}_{X,C}} \prod_{t=0}^{T-1} P_t(\mathbf{c}|\mathbf{x}_0, \ldots, \mathbf{x}_t), \tag{5}$$

where $\mathbf{c}$ is the whole set of recognizable phonetic units or characters plus a blank symbol (modeling confusion information of the speech signal [4]), and the summation is performed over the set of all valid alignments $\mathcal{A}_{X,C}$. From Eq. (5), the acoustic model outputs can be understood as the probability distribution over all the possible label sequences given the sequence of input features $\mathbf{X}$ [46].

The very first attempt to apply CTC to KWS was carried out by Fernández *et al.* [46] using a BiLSTM for acoustic modeling. At training time, this system just needs, along with the training speech signals, the list of training words in order of occurrence. After this first attempt, several works have explored variants of this approach using different RNN architectures like LSTMs [4], [60], [61], [139], BiLSTMs [84], [98] and GRUs [61], [140], as well as considering different phonetic units such as phonemes [60], [84] and Mandarin syllables [8], [139]. In general, these systems are shown to be superior to both LVCSR- and keyword/filler HMM-based KWS systems with less or no additional computational cost [4], [8], [139]. Notice that since CTC requires searching for the keyword phonetic unit sequence on a lattice, this approach is also suitable for open-vocabulary KWS.

2) SEQUENCE-TO-SEQUENCE MODELS

CTC assumes conditional label independence, i.e., past model outputs do not influence current predictions (see Eq. (5)). Hence, in the context of KWS and ASR in general, CTC may need an external language model to perform well. Therefore, a more convenient approach for KWS acoustic modeling might be the use of sequence-to-sequence (Seq2Seq) models, first proposed in [141] for language translation. Figure 7 illustrates an example of Seq2Seq model. In short, Seq2Seq models are comprised of an RNN encoder[7] summarizing the variable-length input sequence

[7]In [9], Shan *et al.* show, for KWS, the superiority of CRNN encoders with respect to GRU ones, which, in turn, are better than LSTM encoders.

into a fixed-dimensional vector followed by an RNN decoder generating a variable-length output sequence conditioned on both the encoder output and past decoder predictions.

Besides for related tasks like QbE KWS [142], Seq2Seq models such as an RNN-Transducer (RNN-T) have also been studied for deep spoken KWS [60], [62], [101], [143]. RNN-T, integrating both acoustic and language models (and predicting phonemes), is able to outperform a CTC KWS system even when the latter exploits an external phoneme N-gram language model [60].

3) THE ATTENTION MECHANISM

As aforementioned, in Seq2Seq models, the encoder has to condense all the needed information into a fixed-dimensional vector regardless the (variable) length of the input sequence, which might be challenging. The attention mechanism [144], similarly to human listening attention, might assist in this context by focusing on the speech sections that are more likely to comprise a keyword [9].

Let $\mathbf{h}_t$ be the hidden state of the RNN encoder of a Seq2Seq model at time step t:

$$\mathbf{h}_t = \text{Encoder}\left(\mathbf{x}_t, \mathbf{h}_{t-1}\right). \tag{6}$$

Before decoding it, the whole input sequence $\mathbf{X} = (\mathbf{x}_0, \ldots, \mathbf{x}_{T-1})$ has to be read, since $\mathbf{h}_{T-1}$ is the fixed-dimensional vector summarizing the whole input sequence that is finally input to the decoder (see Figure 7). To assist the decoder, a context-relevant subset of $\{\mathbf{h}_0, \ldots, \mathbf{h}_{T-1}\}$ can be *attended* to yield $\mathbf{A}$, which is to be used instead of $\mathbf{h}_{T-1}$:

$$\mathbf{A} = \sum_{t=0}^{T-1} \alpha_t \mathbf{h}_t, \tag{7}$$

where $\alpha_t = \text{Attend}\left(\mathbf{h}_t\right)$, being $\text{Attend}\left(\cdot\right)$ an attention function [144] and $\sum_t \alpha_t = 1$.

The integration of an attention mechanism (including a variant called multi-head attention [144]) in (primarily) Seq2Seq acoustic models in order to focus on the keyword(s) of interest has successfully been accomplished by a number of works, e.g., [26], [32], [60], [68], [133], [143], [145]. These works find that incorporating attention provides KWS performance gains with respect to counterpart Seq2Seq models without attention.

Lastly, let us notice that attention has also been studied in conjunction with TDNNs for KWS [12], [16]. Particularly, in [16], thanks to exploiting shared weight self-attention, Bai *et al.* reproduce the performance of the deep residual learning model `res15` of Tang and Lin [30] by using 20 times less parameters, i.e., around 12k parameters only.

D. ACOUSTIC MODEL TRAINING

Once the acoustic model architecture has been designed (see the previous subsections) or optimally "searched" [95], [146], it is time to discriminatively estimate its parameters according to an optimization criterion —defined by a loss function— by means of backpropagation [147] and using labeled/annotated speech data (see Section VIII in the latter respect).

1) LOSS FUNCTIONS

Apart from CTC [63], which has been examined in the previous subsection, cross-entropy loss [73], [74] is, by far, the most popular loss function for training deep spoken KWS acoustic models. For example, cross-entropy loss $\mathcal{L}_{\text{CE}}$ is considered by [12], [16], [22], [29]–[32], [42], [43], [76], [93], [121], [123], [124], and, retaking the notation of Section II, can be expressed as

$$\mathcal{L}_{\text{CE}} = -\sum_i \sum_{n=1}^{N} l_n^{\{i\}} \log\left(\mathbf{y}_n^{\{i\}}\right), \tag{8}$$

where $l_n^{\{i\}}$ is the binary true (training) label corresponding to the input feature segment $\mathbf{X}_{\{i\}}$. Notice that when the acoustic model is intended to produce subword-level posteriors, commonly, training labels are generated by forced alignment using an LVCSR system [22], [31], [42], which will condition the subsequent KWS system performance.

First proposed in [148], max-pooling loss is an alternative to cross-entropy loss that has also been studied for KWS purposes [29], [39], [71]. In the context of KWS, the goal of max-pooling loss is to teach the acoustic model to only trigger at the highest confidence time near the end of the keyword [29]. Let $\hat{\mathbf{L}}$ be the set of all the indices of the input feature segments in a minibatch belonging to any non-keyword class. In addition, let $y_p^\star$ be the largest target posterior corresponding to the p-th keyword sample in the minibatch, where $p = 1, \ldots, P$ and P is the total number of keyword samples in the minibatch. Then, max-pooling loss can be expressed as

$$\mathcal{L}_{\text{MP}} = -\sum_{i \in \hat{\mathbf{L}}} \sum_{n=1}^{N} l_n^{\{i\}} \log\left(\mathbf{y}_n^{\{i\}}\right) - \sum_{p=1}^{P} \log\left(y_p^\star\right). \tag{9}$$

From (9), we can see that max-pooling loss is cross-entropy loss for any non-keyword class (left summand) while, for each keyword sample, the error is backpropagated for a single input feature segment only (right summand). Max-pooling loss has proven to outperform cross-entropy loss in terms of KWS performance, especially when the acoustic model is initialized by cross-entropy loss training [29]. Weakly-constrained and smoothed max-pooling loss variants are proposed in [39] and [71], respectively, which benefit from lowering the dependence on the accuracy of LVCSR forced alignment.

2) OPTIMIZATION PARADIGMS

In deep KWS, the most frequently used optimizers are stochastic gradient descent (SGD) [149] (normally with momentum), e.g., see [30], [31], [49]–[51], [53], [76], [91], [100], [121], [138], [143], [146], and Adam [150], e.g., see [9], [12], [16], [26], [32], [42], [43], [47], [48], [68], [70], [78], [90], [98], [151]. It is also a common practice

to implement a mechanism shrinking the learning rate over epochs [9], [12], [16], [22], [29], [43], [48], [49], [51], [53], [68], [70], [76], [121], [152]. Furthermore, many deep KWS works, e.g., [9], [49]–[51], [90], [100], [143], deploy a form of parameter regularization like weight decay and dropout. While random acoustic model parameter initialization is the normal approach, initialization based on transfer learning from LVCSR acoustic models has proven to lead to better KWS models by, e.g., alleviating overfitting [22], [58], [101].

V. POSTERIOR HANDLING

In order to come up with a final decision about the presence or not of a keyword in an audio stream, the sequence of posteriors yielded by the acoustic model, $\mathbf{y}^{\{i\}}$, needs to be processed. We differentiate between two main posterior handling modes: non-streaming (static) and streaming (dynamic) modes.

A. NON-STREAMING MODE

Non-streaming mode refers to standard multi-class classification of independent input segments comprising a single word each (i.e., isolated word classification). To cover the duration of an entire word, input segments have to be long enough, e.g., around 1 second long [153], [154]. In this mode, commonly, given an input segment $\mathbf{X}_{\{i\}}$, this is assigned to the class with the highest posterior probability as in Eq. (4). This approach is preferred over picking classes yielding posteriors above a sensitivity (decision) threshold to be set, since experience tells [82], [128]–[130] that non-streaming deep KWS systems tend to produce very peaked posterior distributions. This might be attributed to the fact that non-streaming systems do not have to deal with inter-class transition data as in the dynamic case (see the next subsection), but with well-defined, isolated class realizations.

As mentioned in Section II, KWS is not a static task but a dynamic one, which means that a KWS system has to continuously process an input audio stream. Therefore, it is obvious that the non-streaming mode lacks some realism from a practical point of view. Despite this, isolated word classification is considered by a number of deep KWS works, e.g., [16], [30], [32], [48]–[52], [58], [69], [82], [89], [99], [109], [125], [128]–[130]. We believe that this is because of the simpler experimental framework with respect to that of the dynamic or streaming case. Fortunately, non-streaming performance and streaming performance seem to be highly correlated [129], [130], which makes non-streaming KWS research more relevant than it might look at first sight.

B. STREAMING MODE

Streaming mode alludes to the continuous processing (normally in real-time) of an input audio stream in which keywords are not isolated/segmented. Hence, in this mode, any given segment may or may not contain (parts of) a keyword. In this case, the acoustic model yields a time sequence of (raw) posteriors $\left\{\ldots, \mathbf{y}^{\{i-1\}}, \mathbf{y}^{\{i\}}, \mathbf{y}^{\{i+1\}}, \ldots\right\}$ with strong local correlations. Due to this, the sequence of raw posteriors, which is inherently noisy, is typically smoothed over time —e.g., by moving average— on a class basis [15], [22], [29], [42], [43], [45], [56], [58], [72], [76], [77] before further processing.

Let us denote by $\bar{\mathbf{y}}^{\{i\}}$ the smoothed version of the raw posteriors $\mathbf{y}^{\{i\}}$. Furthermore, let us assume that each of the N classes of a deep KWS system represents a whole word (which is a common case). Then, the smoothed word posteriors $\bar{\mathbf{y}}^{\{i\}}$ are often directly used to determine the presence or not of a keyword either by comparing them with a sensitivity threshold[8] [29], [43], [58] or by picking, within a time sliding window, the class with the highest posterior [76]. Notice that since consecutive input segments $\left\{\ldots, \mathbf{X}_{\{i-1\}}, \mathbf{X}_{\{i\}}, \mathbf{X}_{\{i+1\}}, \ldots\right\}$ may cover fragments of the same keyword realization, false alarms may occur as a result of recognizing the same keyword realization multiple times from the smoothed posterior sequence $\left\{\ldots, \bar{\mathbf{y}}^{\{i-1\}}, \bar{\mathbf{y}}^{\{i\}}, \bar{\mathbf{y}}^{\{i+1\}}, \ldots\right\}$. To prevent this problem, a simple, yet effective mechanism consists of forcing the KWS system not to trigger for a short period of time right after a keyword has been spotted [29], [43].

Differently from the above case, let us now consider the two following scenarios:

1) Each of the N classes still represents a whole word but keywords are composed of multiple words (e.g., "OK Google").
2) Each of the N classes represents a subword unit (e.g., a syllable) instead of a whole word.

To tackle such scenarios, the first deep spoken KWS system [22] proposed a simple method processing the smoothed posteriors $\bar{\mathbf{y}}^{\{i\}}$ in order to produce a keyword presence decision. Let us assume that the first class C_1 corresponds to the non-keyword class and that the remaining $N-1$ classes represent subunits of a single keyword.[9] Then, a time sequence of confidence scores $S_c^{\{i\}}$ can be computed as [22]

$$S_c^{\{i\}} = \sqrt[N-1]{\prod_{n=2}^{N} \max_{h_{\max}(i) \leq k \leq i} \bar{\mathbf{y}}_n^{\{k\}}}, \tag{10}$$

where $h_{\max}(i)$ indicates the onset of the time sliding window. A keyword is detected every time $S_c^{\{i\}}$ exceeds a sensitivity threshold to be tuned. This approach has been widely used in the deep KWS literature, e.g., [45], [56], [77].

In [15], Eq. (10) is subject to the constraint that the keyword subunits trigger in the correct order of occurrence within the keyword, which contributes to decreasing false alarms. This improved version of the above posterior handling method is also considered by a number of deep KWS systems, e.g., [42], [72].

When each of the N classes of a deep KWS system represents a subword unit like a syllable or context-independent phoneme, a searchable lattice may be built from the time

[8]This decision threshold might be set by optimizing, on a development set, some kind of figure of merit (see also Section IX on evaluation metrics).

[9]This method can easily be extended to deal with more than one keyword [22].

sequence of posteriors $\mathbf{y}^{\{i\}}$. Actually, this is typically done in the context of CTC [4], [8]. Then, the goal is to find, from the lattice, the most similar subword unit sequence to that of the target keyword. If the score resulting upon the search on the lattice is greater than a predefined score threshold, a keyword is spotted. Notice that this approach, despite its higher complexity, provides a great flexibility by, for example, allowing a user defining her/his own keywords.

VI. ROBUSTNESS IN KEYWORD SPOTTING

Normalizing the effect of acoustic variability factors such as background noise and room reverberation is paramount to assure good KWS performance in real-life conditions. This section is intended to review the scarce literature on KWS robust against, primarily but not only, background noise and far-field conditions. The motivation behind primarily dealing with the two latter acoustic variability factors lies in typical use cases of KWS technology.[10]

This section has been arranged according to a taxonomy that segregates front- and back-end methods, which reflects the available literature on KWS robustness. Let us stress that these are normally cross-cutting methods, since they either come from or can be applied to other areas like ASR.

A. FRONT-END METHODS

Front-end methods refer to those techniques that modify the speech signal before it is fed to the DNN acoustic model. In this subsection, we further differentiate among gain control for far-field conditions, DNN feature enhancement, adaptive noise cancellation and beamforming methods.

1) GAIN CONTROL FOR FAR-FIELD CONDITIONS

Keyword spotting deployment is many times conceived to facilitate real hands-free communication with devices such as smart speakers or in-vehicle systems that are located at a certain distance from the speaker. This means that communication might take place in far-field conditions, and, due to distance attenuation, background noise and reverberation can be particularly harmful.

Prabhavalkar *et al.* [15] were the first to propose the use of automatic gain control (AGC) [155] to provide robustness against background noise and far-field conditions for deep KWS. The philosophy behind AGC is based on selectively amplifying the audio signal depending on whether speech is present or absent. This type of selective amplification is able to yield a significant reduction of miss detections in the far-field scenario [15].

Later, a more popular [61], [93], [94], [122] and simpler AGC method called PCEN (Per-Channel Energy Normalization) [156] was proposed for KWS. Keeping the original notation of [156], $E(t,f)$ represents (Mel) filterbank energy at time frame t and frequency bin f, and

$$M(t,f) = (1-s)M(t-1,f) + sE(t,f) \tag{11}$$

[10]For example, activation of voice assistants typically takes place at home in far-field conditions and with some TV or music background noise.

is a time smoothed version of $E(t,f)$, where $0 < s < 1$ is a smoothing coefficient. Thus, PCEN is intended to replace the typical log compression of filterbank features as follows:

$$\text{PCEN}(t,f) = \left(\frac{E(t,f)}{(\epsilon + M(t,f))^{\alpha}} + \delta\right)^{r} - \delta^{r}, \tag{12}$$

where ϵ prevents division by zero, $\alpha \in (0, 1)$ defines the gain normalization strength, and δ and r determine the root compression. As we can see from Eq. (12), the energy contour of $E(t,f)$ is dynamically normalized by $M(t,f)$ on a frequency band basis, which yields significant KWS performance gains under far-field conditions since $M(t,f)$ mirrors the loudness profile of $E(t,f)$ [156].

An appealing aspect of PCEN is that all its operations are differentiable. As a result, PCEN can be integrated in the DNN acoustic model in order to comfortably tune its set of parameters —i.e., s, ϵ, α, δ and r— towards the optimization of KWS performance during acoustic model training [156].

2) DNN FEATURE ENHANCEMENT

The powerful modeling capabilities of DNNs can also be exploited to clean the noisy speech features (usually, magnitude spectral features) before these are input to the KWS acoustic model. A variety of approaches can be followed:

1) *Enhancement Mask Estimation:* The aim of this approach is to estimate, from the noisy observation (e.g., noisy Mel spectra [157]) and using a neural network (e.g., a CRNN [157]), a multiplicative denoising time-frequency mask to be applied to the noisy observation [157], [158]. The result is then passed to the acoustic model.
2) *Noise Estimation:* A DNN (e.g., a CNN with dilated convolutions and residual connections [159]) might also be used to provide an estimate of the distortion that contaminates the target speech signal. The estimated distortion can then be subtracted from the noisy observation before feeding the acoustic model with it [159].
3) *Clean Speech Estimation:* In this case, the DNN front-end directly produces an estimate of the clean speech features, from the noisy observation, to be input to the acoustic model. While this approach has been studied for robust ASR [160], to the best of our knowledge and surprisingly, this has not been the case for KWS.
4) *Filter Parameter Estimation:* The parameters of an enhancement filter (e.g., a Wiener filter [158]) to be applied to the noisy observation before further processing can be estimated by means of a DNN. Similarly to the above case, while this has been studied for robust ASR [158], this has not been the case for KWS.

Regardless of the chosen approach, the DNN front-end and the KWS acoustic model can be jointly trained following a multi-task learning scheme to account the complementary objectives of the front-end and the acoustic model. By making the DNN front-end aware of the global keyword detection goal [157], [159], superior KWS performance can be

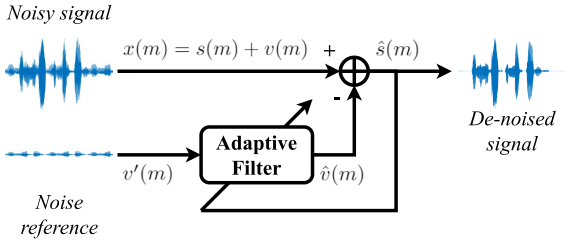

FIGURE 8. Block diagram of adaptive noise cancellation. A signal of interest $s(m)$ is retrieved from a noisy observation $x(m) = s(m) + v(m)$ by subtracting an estimate of $v(m)$, $\hat{v}(m)$. This estimate is obtained by filtering a noise reference $v'(m)$ that originates from the same noise source as $v(m)$ (i.e., $v(m)$ and $v'(m)$ are highly correlated). The filter weights are continuously adapted to typically minimize the power of the estimated signal of interest $\hat{s}(m)$.

achieved in comparison with independent training of the two components.

One conclusion is that, oddly, DNN feature enhancement is a rather unexplored area in the context of KWS. This contrasts with the case of robust ASR, which has widely and successfully studied the application of this type of de-noising front-ends [158], [160]. Immediate future work on robust KWS could address this imbalance, especially by exploring promising time domain solutions that can benefit from phase information [161].

3) ADAPTIVE NOISE CANCELLATION

Presumably thinking of voice assistant use cases, Google developed a series of noise-robust KWS methods based on dual-microphone adaptive noise cancellation (ANC) to particularly deal with speech interference [94], [122], [162]. The working principle of ANC is outlined in Figure 8. The reason for accounting a dual-microphone scenario is that Google's smart speaker Google Home has two microphones [163]. It is interesting to point out that the authors of this series of ANC works also tried to apply beamforming and multi-channel Wiener filtering,[11] but they only found marginal performance gains by doing so [94].

In [122], Google researchers proposed a de-noising front-end inspired by the human auditory system. In short, the de-noising front-end works by exploiting posterior probability feedback from the KWS acoustic model:

1) If the acoustic model finds that voice is absent, the weights of a recursive least squares (RLS) ANC filter working in the short-time Fourier transform (STFT) domain are updated;
2) If the posterior probabilities computed by the acoustic model are inconclusive (i.e., the presence of a keyword is uncertain), the most recent ANC weights are used to filter/clean the input signal and the presence of a keyword is rechecked.

[11]Notice that multi-channel Wiener filtering is equivalent to minimum variance distortionless response (MVDR) beamforming followed by single-channel Wiener post-filtering [164], [165].

In a similar vein, a so-called hotword cleaner was reported in [94], which overcomes one of the shortcomings of the above ANC approach [122]: the increased latency and CPU usage derived from having to run the acoustic model twice (one to provide feedback to the de-noising front-end and another for KWS itself). The hotword cleaner [94] leverages the following two characteristics of the KWS scenario to deploy a simple, yet effective de-noising method: *1)* there is typically no speech just before a keyword, and *2)* keywords are of short duration. Bearing these two characteristics in mind, the hotword cleaner [94] simply works by continuously computing fast-RLS ANC weights that are stored and applied to the input signal with a certain delay to clean and not damage the keyword. This methodology was generalized to an arbitrary number of microphones in [162]. Overall, all of these ANC-based methods bring significant KWS performance improvements in everyday noisy conditions that include strong TV and music background noise.

4) BEAMFORMING

Spatial filtering, also known as beamforming, enables the exploitation of spatial cues in addition to time and frequency information to boost speech enhancement quality [166]. Similarly to the aforementioned case with DNN feature enhancement, KWS lags several steps behind ASR regarding the integration of beamforming as done, e.g., in [167].

To the best of our knowledge, [97] is the first research studying beamforming for deep KWS. In particular, [97] applies four *fixed* beamformers that are arranged to uniformly sample the horizontal plane. The acoustic model is then fed with the four resulting beamformed signals plus a reference signal picked from one of the array microphones to avoid degrading performance at higher signal-to-noise ratios (SNRs) [97]. The acoustic model incorporates an attention mechanism [144] to weigh the five input signals, which can be thought as a steering mechanism pointing the effective beam towards the target speaker. Actually, the motivation behind using fixed beamformers lies in the difficulty of estimating the target direction in noisy conditions. However, notice that the attention mechanism implicitly estimates it.

The same authors of [97] went farther in [96] by replacing the set of fixed beamformers by a set of data-dependent, multiplicative spectral masks playing an equivalent role. The latter masks, which are estimated by a neural network, can be interpreted as *semi-fixed* beamformers. This is because though they are data-dependent, mask look directions (equivalent to look directions of beamformers) are still fixed. This beamforming front-end, which is trained jointly with the acoustic model, outperforms the previous fixed beamforming approach, especially at lower signal-to-interference ratios (SIRs) [96].

There is still a long way to go regarding the application of beamforming to deep KWS. More specifically, despite the aforementioned steering role of the attention mechanism, we believe that deep beamforming that does not pre-arrange

the look direction but estimates it continuously based on microphone signals —as in, e.g., [168]— is worth to explore.

B. BACK-END METHODS

Back-end methods refer to techniques applied within the acoustic model to primarily improve its generalization ability to a variety of acoustic conditions. The rest of this subsection is devoted to discuss the following matters: multi-style and adversarial training, robustness to keyword data scarcity, the class-imbalance problem and other back-end methods.

1) MULTI-STYLE TRAINING

One of the most popular and effective back-end methods to, especially, deal with background noise and reverberation is multi-style training of the KWS acoustic model (see, e.g., [15], [32], [39], [43], [50], [60], [71], [88], [90], [118], [169], [170]). Multi-style training, which has some regularization effect preventing overfitting [118], simply consists of training the acoustic model with speech data contaminated by a variety of distortions trying to better reflect what is expected to be found at test time.

Usually, distorted speech data are generated by contaminating —e.g., by background noise addition at different SNR levels— clean speech data in an artificial manner (see Section VIII for practical details). This artificial distortion procedure is known as *data augmentation* [171]. For instance, a series of data augmentation policies like time and frequency masking is defined by a tool like SpecAugment [172]. First proposed for end-to-end ASR, SpecAugment has recently become a popular way for generating distorted speech data, also for KWS training purposes [32], [39], [90], [100], [143], [151].

2) ADVERSARIAL TRAINING

Deep neural networks often raise the following issue: networks' outputs might not be smooth with respect to inputs [173], e.g., because of the lack of enough training data. This might involve, for example, that a keyword correctly classified by the acoustic model is misclassified when a very small perturbation is added to such a keyword. This kind of subtly distorted input to the network is what we call an *adversarial example*. Interestingly, adversarial examples can be generated by means of techniques like the fast gradient sign method (FGSM) [174] to re-train with them a well-trained KWS acoustic model. The goal of this is to improve robustness by smoothing the distribution of the acoustic model. This approach, which can be interpreted as a type of data augmentation, has shown to be effective to drastically decrease false alarms and miss detections for an attention-based Seq2Seq acoustic model [26]. Alternatively, [45] proposes to replace, with the same goal, adversarial example re-training by adversarial regularization in the loss function. Wang *et al.* [45] demonstrate that the latter outperforms the former under far-field and noisy conditions when using a DS-CNN acoustic model for KWS.

3) ROBUSTNESS TO KEYWORD DATA SCARCITY

To effectively train a KWS acoustic model, a sufficient amount of speech data is required. This normally includes a substantial number of examples of the specific keyword(s) to be recognized. However, there is a number of possible reasons for which we might suffer from keyword data scarcity. Certainly, collecting additional keyword samples can help to overcome the problem. Nevertheless, speech data collection can be costly and time-consuming, and is often infeasible. Instead, a smart way to obtain additional keyword samples for model training is by synthetically generating them through text-to-speech technology. This type of data augmentation has proven to be highly effective by significantly improving KWS performance in low-resource keyword settings [62], [175], [176]. In particular, in [62], it is found that it is important that synthetic speech reflects a wide variety of tones of voice (i.e., speaker diversity) for good KWS performance.

4) THE CLASS-IMBALANCE PROBLEM

The class-imbalance problem refers to the fact that, typically, many more non-keyword than keyword samples are available for KWS acoustic model training. Actually, the class-imbalance problem can be understood as a relative keyword data scarcity problem: for obvious reasons, it is almost always easier to access a plethora of non-keyword than keyword samples. The issue lies in that class imbalance can lead to under-training of the keyword class with respect to the non-keyword one.

To reach class balance for acoustic model training, one can imagine many different things that can be done based on data augmentation:

1) Generation of adversarial examples yielding miss detections, e.g., through FGSM [174], to re-train the acoustic model in a class-balanced way;
2) Generation of additional synthetic keyword samples by means of text-to-speech technology [62], [175].

To the best of our knowledge, the above two data augmentation approaches have not been studied for tackling the class-imbalance problem.

Differently, a series of works has proposed to essentially focus on challenging non-keyword samples[12] at training time instead of fully exploiting all the non-keyword samples available [39], [42], [177]. For instance, Liu *et al.* [42] suggested to weigh cross-entropy loss $\mathcal{L}_{\text{CE}}$ (see Eq. (8)) by $\left(1-\mathbf{y}_n^{\{i\}}\right)^{\gamma}$ to come up with focal loss $\mathcal{L}_{\text{FL}}$:

$$\mathcal{L}_{\text{FL}} = -\sum_i \sum_{n=1}^{N} \left(1-\mathbf{y}_n^{\{i\}}\right)^{\gamma} l_n^{\{i\}} \log\left(\mathbf{y}_n^{\{i\}}\right), \qquad (13)$$

where γ is a tunable focusing parameter. As one can easily reason, weighing cross-entropy loss as in Eq. (13) helps to focus training on challenging samples. While this weighting procedure is more effective than regular cross-entropy in

[12] A challenging non-keyword sample can be, e.g., one exhibiting similarities with the keyword in terms of phonetics.

class-imbalanced scenarios [42], notice that it might be able to strengthen the model in a wide sense. Because focal loss $\mathcal{L}_{\text{FL}}$ operates on a frame basis, [177] improved it by also considering the time context when computing the weight for cross-entropy loss. Particularly, such an improvement is equivalent to assigning bigger weights to those frames belonging to non-keyword samples yielding false alarms.

An alternative approach —so-called regional hard-example mining— for dealing with the class-imbalance problem was described in [39]. Regional hard-example mining subsamples the available non-keyword training data to keep a certain balance between keyword and non-keyword samples. Non-keyword sample mining is based on the selection of the most difficult non-keyword samples in the sense that they yield the highest keyword posteriors.

5) OTHER BACK-END METHODS

A few other methods for robustness purposes not falling into any of the above categories can be found in the literature. For instance, [72] extracts embeddings characterizing the acoustic environment that are passed to the acoustic model to carry out KWS which is robust to far-field and noisy conditions. In this way, by making the acoustic model *aware* of the acoustic environment, better keyword prediction can be achieved.

We also recently contributed to noise-robust KWS in [130], where we proposed to interpret every typical KWS acoustic model as the concatenation of a keyword embedding extractor followed by a linear classifier consisting of the typical final fully-connected layer with softmax activation for word classification (see Section II). The goal is to, first, multi-style train the keyword embedding extractor by means of a $(C_{N,2}+1)$-pair loss function extending the idea behind tuple-based losses like N-pair [178] and triplet [179] losses (the latter used both standalone [103] and combined with the reversed triplet and hinge losses [56] for keyword embedding learning). In comparison with these and similar losses also employed for word embedding learning (e.g., a prototypical loss angular variant [180]), in [130], we demonstrate that the $(C_{N,2}+1)$-pair loss reaches larger inter-class and smaller intra-class embedding variation.[13] Secondly, the final fully-connected layer with softmax activation is trained by multi-style keyword embeddings employing cross-entropy loss. This two-stage training strategy is much more effective than standard end-to-end multi-style training when facing unseen noises [130]. Moreover, another appealing feature of this two-stage training strategy is that it increases neither the number of parameters nor the number of multiplications of the model.

VII. APPLICATIONS

Keyword spotting technology (including deep KWS) has a number of applications, which range from the more traditional ones like voice-dialing, interaction with a call center and speech retrieval to nowadays flagship application, namely, the activation of voice assistants.

In addition to the above, KWS technology could be useful, e.g., to assist disabled people like vision-impaired pedestrians when it comes to the activation of pedestrian call buttons in crosswalks. For example, [87] proposes the use of a CRNN-based KWS system [93] for the activation of pedestrian call buttons via voice, thereby contributing to improve accessibility in public areas to people with the above-mentioned disability.

In-vehicle systems can also benefit from voice control. For example, in [77], Tan *et al.* explore multi-source fusion exploiting variations of vehicle's speed and direction for *online* sensitivity threshold selection. The authors of [77] demonstrate that this strategy improves KWS accuracy with respect to using a fixed, predetermined sensitivity threshold for the posteriors yielded by the DNN acoustic model.

Moreover, it is worth noticing that KWS is a technology that is sometimes better suited than ASR to the solution of certain problems where the latter is typically employed. This is the case, for instance, of by-topic audio classification and audio sentiment detection [181], [182], since the accuracy of these tasks rather relies on being able to correctly spot a very focused (i.e., quite limited) vocabulary in the utterances. In other words, lexical evidence is sparse for such tasks.

Some work has explored KWS also for voice control of videogames [138], [152]. Particularly, [138] points out how KWS becomes an extremely difficult task when it comes to dealing with children controlling videogames with their voice due to excitement and, generally speaking, the nature of children and children's voice [183]. To partially deal with this, the authors of [138] propose the detection of overlapping keywords in the context of a multiplayer side-scroller game called Mole Madness. Since BiLSTMs have proven to work well for children's speech [184], a BiLSTM acoustic model with 2^N output classes —where N is the number of keywords— is used to represent all possible combinations of overlapping keywords. It is found that, under the videogame conditions, modeling the large variations of children's speech time structure is challenging even for a relatively large BiLSTM.

Other KWS applications include voice control of home automation [185], even the navigation of complex procedures in the International Space Station [186], etc.

A. PERSONALIZED KEYWORD SPOTTING SYSTEMS

For some of the above applications, having a certain degree of personalization in the sense that only a specific user is allowed to utilize the KWS system can be a desirable feature. Towards this personalization goal, some research has studied the combination of KWS and speaker verification [10], [76], [140], [159]. While [10], [140] employ independently trained deep learning models to perform both tasks, [76], [159] address, following a multi-task learning scheme, joint KWS and speaker verification with contradictory conclusions,

[13] This is because the $(C_{N,2}+1)$-pair loss constrains the way the training samples belonging to different classes relate to each other in terms of embedding distance.

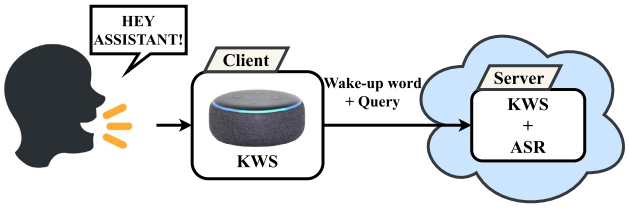

FIGURE 9. Typical voice assistant client-server framework.

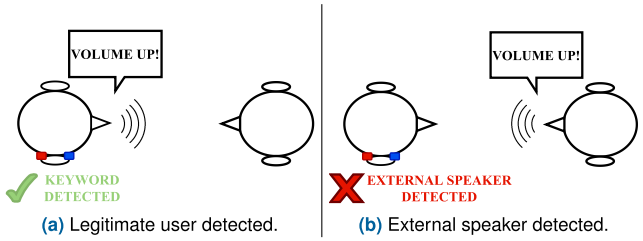

FIGURE 10. Users' own voice/external speaker detection in the context of voice control of hearing aids. Red and blue dots symbolize the two microphones of a hearing aid sitting behind the ear.

since KWS performance is negatively and positively affected in [76] and [159], respectively, by the integration of speaker verification. A reason for this could be that, unlike in [159], higher-level features are shared for both tasks in [76], so this further preservation of speaker information may contaminate the phonetic information required to carry out KWS.

Personalization can be of particular interest for voice activation of voice assistants [187] as well as for voice control of hearing assistive devices like hearing aids. These two KWS applications are reviewed in a bit more detail in the next subsections.

B. VOICE ACTIVATION OF VOICE ASSISTANTS

The flagship application of (deep) KWS is the activation of voice assistants like Amazon's Alexa, Apple's Siri, Google's Assistant and Microsoft's Cortana. Actually, without fear of error, we can say that revitalization of KWS research over the last years is owed to this application [28]. And there is a compelling reason for this: forecasts suggest that, by 2024, the number of voice assistant units will exceed that of world's population [188].

Figure 9 illustrates the typical voice assistant client-server framework. The client consists of an electronic device like a smartwatch or a smart speaker integrating the client-side of a voice assistant and an always-on KWS system to detect when a user wakes up the assistant by uttering a trigger word/phrase, e.g., "hey assistant!". To limit the impact on the battery life, the KWS system has to be necessarily light. In this vein, Apple employs a two-pass detection strategy [187]. By this, a very light, always-on KWS system listens for the corresponding wake-up word. If this is detected, a more complex and accurate KWS system —also placed on the client device— is used to double check whether or not the wake-up word has been really uttered.

When the wake-up word is spotted on the client-side, the supposed wake-up word audio and subsequent query audio are sent to a server on which, first, the presence of the wake-up word is checked for a second or third time by using much more powerful and robust LVCSR-based KWS [2], [187], [189]. If, finally, the LVCSR-based KWS system determines that the wake-up word is not present, the subsequent audio is discarded and the process is ended. Otherwise, ASR is applied to the supposed query audio and the result is further processed —e.g., using natural language processing techniques— to provide the client device with a response. In the context of Google's Assistant [2], it is shown that this server-side wake-up word check drastically reduces the false alarm rate while marginally increasing the rate of miss detections. Notice that this server-side check could be useful for mitigating privacy issues as a result of pseudo-query audio leakage if it were not for the fact that the supposed wake-up word audio and query audio are inseparably streamed to the server. Interestingly, Garg *et al.* [190] have recently proposed a streaming Transformer encoder carrying out the double check efficiently on the client-side, which can truly help to mitigate privacy issues.

C. VOICE CONTROL OF HEARING ASSISTIVE DEVICES

Manually operating small, body-worn devices like hearing aids is not always feasible or can be cumbersome. One reason could be that hands are busy doing other activities like cooking or driving. Another cause could be that the wearer is an elderly person with reduced fine motor skills. Whatever the reason is, KWS can help to deploy voice interfaces to comfortably operate such a kind of devices. Furthermore, these devices are personal devices, so it is desirable that the user is the only person who can handle them.

In the above respect, in [128], [129] we studied an alternative way to speaker verification to provide robustness against external speakers (i.e., personalization) in KWS for hearing aids as exemplified by Figure 10. Particularly, we extended the deep residual learning model proposed by Tang and Lin [30] to jointly perform KWS and users' own voice/external speaker detection following a multi-task learning scheme. A keyword prediction is then taken into account if and only if the multi-task network determines that the spotted keyword was uttered by the legitimate user. Thanks to exploiting GCC-PHAT (Generalized Cross-Correlation with PHAse Transform) [191] coefficients from dual-microphone hearing aids in the perceptually-motivated constant-Q transform [192] domain, we achieve almost flawless users' own voice/external speaker detection [129]. This is because phase difference information is extremely useful to characterize the virtually time-invariant position of the user's mouth with respect to that of the hearing aid. It is worth noting that this experimental validation was carried out on a hearing aid speech database created by convolving the Google Speech Commands Dataset *v2* [154] with acoustic transfer functions measured in a hearing aids set-up.

TABLE 1. A selection of the most significant speech datasets employed for training and validating deep KWS systems. "P.A." stands for "publicly available", while "Y" and "N" mean "yes" and "no", respectively. Furthermore, "+ *sampl.*" ("- *sampl.*") refers to the size of the positive/keyword (negative/non-keyword) subset, and "*Size*" denotes the magnitude of the whole set. Such sizes are given, depending on the available information, in terms of either the number of samples or time length in hours (h). Unknown information is indicated by hyphens.

Ref.	Name	Developer	P.A.?	Language	Noisy?	No. of KW	Training set			Test set		
							Size	*+ sampl.*	*- sampl.*	*Size*	*+ sampl.*	*- sampl.*
[202]	-	Alibaba	N	Mandarin	Y	1	24k h	-	-	-	12k	600 h
[93]	-	Baidu	N	English	Y	1	12k	-	-	2k	-	-
[42]	-	Chinese Academy of Sciences	N	Mandarin	Y	2	47.8k	8.8k	39k	-	1.7k	-
[58]	-	Fluent.ai	N	English	Y	1	50 h	5.9k	-	22 h	1.6k	-
[22]	-	Google	N	English	Y	10	>3k h	60.7k	133k	81.2k	11.2k	70k
[28]	-	Google	N	English	Y	14	326.8k	10k	316.8k	61.3k	1.9k	59.4k
[61]	-	Harbin Institute of Technology	N	Mandarin	-	1	115.2k	19.2k	96k	28.8k	4.8k	24k
[103]	-	Logitech	N	English	-	14	-	-	-	-	-	-
[26]	-	Mobvoi	N	Mandarin	Y	1	67 h	20k	54k	7 h	2k	5.9k
[169]	-	Sonos	Y	English	Y	16	0	0	0	1.1k	1.1k	0
[96]	-	Tencent	N	Mandarin	Y	1	339 h	224k	100k	-	-	-
[45]	-	Tencent	N	Mandarin	Y	1	65.9 h	6.9 h	59 h	8.7 h	0.9 h	7.8 h
[56]	-	Tencent	N	Mandarin	Y	42	22.2k	15.4k	6.8k	10.8k	7.4k	3.4k
[9]	-	Xiaomi	N	Mandarin	-	1	1.7k h	188.9k	1M	52.2 h	28.8k	32.8k
[199]	AISHELL-2 (13)	AISHELL	Y	Mandarin	N	13	24.8 h	>24k	-	16.7 h	>8.4k	-
[199]	AISHELL-2 (20)	AISHELL	Y	Mandarin	N	20	35 h	>34k	-	23.9 h	>12k	-
[108]	"Alexa"	Amazon	N	English	Y	1	495 h	-	-	100 h	-	-
[153]	Google Speech Commands Dataset *v1*	Google	Y	English	Y	10	51.7k	18.9k	32.8k	6.5k	2.4k	4.1k
[154]	Google Speech Commands Dataset *v2*	Google	Y	English	Y	10	84.6k	30.8k	53.8k	10.6k	3.9k	6.7k
[59]	"Hey Siri"	Apple	N	English	Y	1	500k	250k	250k	-	6.5k	2.7k h
[200]	Hey Snapdragon Keyword Dataset	Qualcomm	Y	English	N	4	-	-	-	4.3k	4.3k	-
[78]	Hey Snips	Snips	Y	English	Y	1	50.5 h	5.9k	45.3k	23.1 h	2.6k	20.8k
[152]	"Narc Ya"	Netmarble	N	Korean	Y	1	130k	50k	80k	800	400	400
[31]	"Ok/Hey Google"	Google	N	English	Y	2	-	1M	-	>3k h	434k	213k
[122]	"Ok/Hey Google"	Google	N	English	Y	2	-	-	-	247 h	4.8k	7.5k
[17]	Ticmini2	Mobvoi	N	Mandarin	Y	2	157.5k	43.6k	113.9k	72.9k	21.3k	51.6k

VIII. DATASETS

Data are an essential ingredient of any machine learning system for both training the parameters of the algorithm (primarily, in our context, the acoustic model parameters) and validating it. Some well-known speech corpora that have been extensively used over the years in the field of ASR are now also being employed for the development of deep KWS systems. For example, LibriSpeech [193] has been used by [55], [76], [142], [151], [169], TIDIGITS [194], by [140], TIMIT [195], by [41], [84], [117], [118], [196], and the Wall Street Journal (WSJ) corpus [197], by [4], [76], [103]. The main problem with these speech corpora is that they were not developed for KWS, and, therefore, they do not standardize a way of utilization facilitating KWS technology reproducibility and comparison. By contrast, KWS research work exploiting these corpora employs them in a variety of ways, which is even reflected by, e.g., the set of considered keywords.

In the following we focus on those datasets particularly intended for KWS research and development, which, normally, are comprised of hundreds or thousands of different speakers who do not overlap across sets (i.e., training, development and test sets), e.g., [17], [26], [56], [61], [68], [78], [93], [102], [154], [198]. Table 1 shows a wide selection of the most significant speech corpora available for training and testing deep KWS systems. From this table, the first inference that we can draw is that the advancement of the KWS technology is led by the private sector of the United States of America (USA) and China. Seven and five out of the seventeen different dataset developers included in Table 1 are, respectively, North American and Chinese corporations. Actually, except for the "Narc Ya" corpus [152], which is in Korean, all the datasets shown in this table are in either English or Mandarin Chinese.

A problem with the above is that the majority of the speech corpora of interest for KWS research and development are not publicly available (P.A.), but they are for (company) internal use only. On many occasions, these datasets are collected by companies to improve their wake-up word detection systems for voice assistants running on smart speakers.

For example, this is the case for the speech corpora reported in [26], [122] and [9], which were collected, respectively, from Mobvoi's TicKasa Fox, Google's Google Home and Xiaomi's AI Speaker smart speakers. Unfortunately, only seven out of twenty six datasets in Table 1 are publicly available: one from Sonos [169], two different arrangements of AISHELL-2 [199] (used in [98]), the Google Speech Commands Dataset *v1* [153] and *v2* [154], the Hey Snapdragon Keyword Dataset [200], and Hey Snips [78], [198] (also used in, e.g., [53], [177]). In case of interest in getting access to any of these speech corpora, the reader is pointed to the corresponding references indicated in Table 1. Among these publicly available datasets, the Google Speech Commands Dataset (*v1* and *v2*) is, by far, the most popular, and has become the *de facto* open reference for KWS development and evaluation. Because of this, the KWS performance comparison presented in Section X is carried out among KWS systems that are evaluated on the Google Speech Commands Dataset. Further information on this corpus is provided in Subsection VIII-A.

Also from Table 1, we can observe that the great majority of datasets are noisy, which means that speech signals are distorted in different ways, e.g., by natural and realistic background acoustic noise or room acoustics. This is generally a must if we want to minimize the mismatch between the KWS performance at the lab phase and that one observable in the inherently-noisy real-life conditions. In particular, dataset acoustic conditions should be as close as possible as those that we expect to find when deploying KWS systems in real-life [201]. Noisy corpora can be classified as natural and/or simulated noisy speech:

1) *Natural Noisy Speech:* Some of the datasets in Table 1 (e.g., [17], [31], [45], [56], [59], [122], [152], [202]) were partially or totally created from natural noisy speech recorded —many times in far-field conditions— by electronic devices such as smart speakers, smartphones and tablets. Often, recording scenarios consist of home environments with background music or TV sound, since this is the target scenario of many KWS systems.
2) *Simulated Noisy Speech:* Some other noisy datasets conceived for KWS —e.g., [15], [22], [28], [31], [42], [58], [93]— were partially or totally generated by artificially distorting clean speech signals through a procedure called data augmentation [171]. Typically, given a clean speech signal, noisy copies of it are created by adding different types of background noises (e.g., daily life noises like babble, café, car, music and street noises) in such a manner that the resulting SNR levels (commonly, within the range $[-5,\ 20]$ dB) are under control. Filtering and Noise-adding Tool (FaNT) [203] is a useful software to create such noisy copies. For example, FaNT was employed in [43], [130] to generate, in a controlled manner, noisier versions of the already noisy Google Speech Commands Dataset. Normally, background noises for data augmentation come from publicly available databases like TUT [204], DEMAND [205], MUSAN [206], NOISEX-92 [207] and CHiME [208], [209]. In addition, alteration of room acoustics, e.g., to simulate far-field conditions from close-talk speech [93], is another relevant data augmentation strategy.

TABLE 2. List of the words included in the Google Speech Commands Dataset *v1* (first six rows) and *v2* (all the rows). Words are broken down by the standardized 10 keywords (first two rows) and non-keywords (last five rows).

Version 1 (*v1*)	Version 2 (*v2*)	yes	no	up	down	left	KW
		right	on	off	stop	go	
		zero	one	two	three	four	Non-KW
		five	six	seven	eight	nine	
		bed	bird	cat	dog	happy	
		house	Marvin	Sheila	tree	wow	
		backward	forward	follow	learn	visual	

Collecting a good amount of natural noisy speech data in the desired acoustic conditions is not always feasible. In such cases, simulation of noisy speech is a smart and cheaper alternative allowing us for obtaining similar technology performance [210].

We can clearly see from Table 1 that the number of keywords per dataset is mostly 1 or 2. A reason for this is that datasets mainly fit the application of KWS that, lately, is boosting research on this technology: wake-up word detection for voice assistants.

Finally, the right part of Table 1 tells some information about the sizes of the training and test sets[14] of the different corpora in terms of either the number of samples (i.e., words, normally) or time length in hours (h) —depending on the available information—. Specifically, "+ *sampl.*" ("- *sampl.*") refers to the size of the positive/keyword (negative/non-keyword) subset, and "*Size*" denotes the magnitude of the whole set. Unknown information is indicated by hyphens. From this table, we note that, as a trend, publicly available datasets tend to be smaller than in-house ones. Furthermore, while the ratio between the sizes of the training and test sets is greater than 1 in all the reported cases except [169], ratio values tend to differ from one corpus to another. Also, *mainly*, the ratio between the sizes of the corresponding negative/non-keyword and positive/keyword subsets is greater than 1, that is, $\frac{\text{- sampl.}}{\text{+ sampl.}} > 1$. This is purposely done to accurately reflect potential scenarios of use consisting of always-on KWS applications like wake-up word detection, in which KWS systems, most of the time, will be exposed to other types of words instead of keywords.

A. GOOGLE SPEECH COMMANDS DATASET

The publicly available Google Speech Commands Dataset [153], [154] has become the *de facto* open benchmark

[14]Many of these corpora also include a development set. However, this part has been omitted for the sake of clarity.

for (deep) KWS development and evaluation. This crowd-sourced database was captured at a sampling rate of 16 kHz by means of phone and laptop microphones, being, to some extent, noisy. Its first version, *v1* [153], was released in August 2017 under a Creative Commons BY 4.0 license [211]. Recorded by 1,881 speakers, this first version consists of 64,727 one-second (or less) long speech segments covering one word each out of 30 possible different words. The main difference between the first version and the second version —which was made publicly available in 2018— is that the latter incorporates 5 more words (i.e., a total of 35 words), more speech segments, 105,829, and more speakers, 2,618. Table 2 lists the words included in the Google Speech Commands Dataset *v1* (first six rows) and *v2* (all the rows). In this table, words are broken down by the standardized 10 keywords (first two rows) and non-keywords (last five rows). To facilitate KWS technology reproducibility and comparison, this benchmark also standardizes the training, development and test sets, as well as other crucial aspects of the experimental framework, including a training data augmentation procedure involving background noises (see, e.g., [30] for further details). Multiple recent deep KWS works have employed either the first version [16], [30], [32], [43], [48]–[52], [57], [58], [67], [69], [70], [86], [90], [100], [125] or the second version [32], [47], [48], [53], [70], [82], [89], [90], [99], [100], [109], [128]–[130], [159], [175] of the Google Speech Commands Dataset.

Despite how valuable this open reference is for KWS research and development, we can raise two relevant points of criticism:

1) *Class Balancing:* The different keyword and non-keyword classes are rather balanced (i.e., they appear with comparable frequencies) in this benchmark, which, as we know, is generally not realistic. See Subsection IX-A for further comments on this question.
2) *Non-Streaming Mode:* Most of the above-referred works using the Google Speech Commands Dataset performs, due to the nature of this corpus, KWS evaluations in non-streaming mode, namely, multi-class classification of independent short input segments. In this mode, a full keyword or non-keyword is surely present within every segment. However, real-life KWS involves the continuous processing of an input audio stream.

A few deep KWS research works [43], [58], [129], [130] have proposed to overcome the above two limitations by generating more realistic streaming versions of the Google Speech Commands Dataset by concatenation of one-second long utterances in such a manner that the resulting word class distribution is unbalanced. Even though the author of the Google Speech Commands Dataset reports some streaming evaluations in the database description manuscript [154], still, we think that this point should be standardized for the sake of reproducibility and comparison, thereby enhancing the usefulness of this valuable corpus.

Lastly, we wish to draw attention to the fact that we produced three outcomes revolving around the Google Speech Commands Dataset *v2*: *1)* a variant of it emulating hearing aids as a capturing device (employed, as mentioned in Subsection VII-C, for KWS for hearing assistive devices robust to external speakers) [128], [129], *2)* another noisier variant with a diversity of noisy conditions[15] (i.e., types of noise and SNR levels) [130], and *3)* manually-annotated speaker gender labels.[16]

IX. EVALUATION METRICS

Obviously, the gold plate test of any speech communication system is a test with relevant end-users. However, such tests tend to be costly and time-consuming. Instead (or in addition to subjective tests), one adheres to objective performance metrics for estimating system performance. It is important to choose a meaningful objective evaluation metric that allows us to determine the goodness of a system and is highly correlated to the subjective user experience. In what follows, we review and provide some criticism of the most common metrics considered in the field of KWS. These metrics are rather intended for binary classification —e.g., keyword/non-keyword— tasks. In the event of having multiple keywords, a common approach consists of applying the metric computation for every keyword and, then, the result is averaged, e.g., see [30], [129], [130].

A. ACCURACY

Accuracy can be defined as the ratio between the number of correct predictions/classifications and the total number of them [212]. In the context of binary classification (e.g., keyword/non-keyword), accuracy can also be expressed from the number of true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN) as follows [213]:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{14}$$

Accuracy $\in [0, 1]$, where 0 and 1 indicate, respectively, worst and perfect classification.

It is reasonable to expect that, in real-life applications like wake-up word detection, KWS systems will hear other word types rather than keywords most of the time. In other words, KWS is a task in which, in principle, the keyword and non-keyword classes are quite unbalanced. Under these circumstances, accuracy tends to be an unsuitable evaluation metric yielding potentially misleading conclusions [214], [215]. Let us illustrate this statement with the following example. Let us consider two different KWS systems *SYS1* and *SYS2*. While *SYS1* is a relatively decent system, *SYS2* is a totally useless one, since it always outputs ''non-keyword'' regardless of the input. Figure 11 depicts, along with an example ground truth sequence, the sequences of keywords (**KW**)

[15]Tools to create this noisy dataset can be freely downloaded from http://ilopez.es.mialias.net/misc/NoisyGSCD.zip

[16]These labels are publicly available at https://ilopezes.files.wordpress.icom/2019/10/gscd_spk_gender.zip

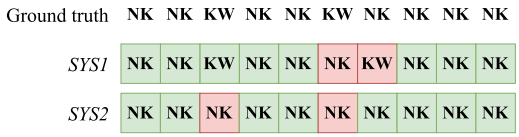

FIGURE 11. Example of two different KWS systems *SYS1* and *SYS2* recognizing a sequence of keywords (KW) and non-keywords (NK). The ground truth sequence is also shown on top.

and non-keywords (**NK**) predicted by *SYS1* and *SYS2*. In this situation, both KWS systems perform with 80% accuracy, even though *SYS2* is useless while *SYS1* is not. Thus, particularly in unbalanced situations, more appropriate evaluation metrics than accuracy may be required, and these are discussed in the next subsections.

In spite of its disadvantage in unbalanced situations, accuracy is a widely used evaluation metric for deep KWS, especially when performing evaluations on the popular Google Speech Commands Dataset [153], [154] in non-streaming mode [16], [30], [32], [48]–[52], [58], [69], [89], [91], [99], [109], [125]. In this latter case, accuracy can still be considered a meaningful metric, since the different word classes are rather balanced in the Google Speech Commands Dataset benchmark. Hence, the main criticism that might be raised here is the lack of realism of the benchmark itself, as discussed in Subsection VIII-A. Nevertheless, we have experimentally observed for KWS a strong correlation between accuracy on a quite balanced scenario and more suitable metrics like F-score (see Subsection IX-C) on a more realistic, unbalanced scenario [129], [130]. This might suggest that the employment of accuracy, although not ideal, can still be useful under certain experimental conditions to adequately explain the goodness of KWS systems.

B. RECEIVER OPERATING CHARACTERISTIC AND DETECTION ERROR TRADE-OFF CURVES

Let TPR denote the true positive rate —also known as recall [216]—, which is defined as the ratio

$$\text{TPR} = \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{15}$$

Notice that Eq. (15) is the probability that a positive sample (i.e., a keyword in this paper) is correctly detected as such. Similarly, let FPR be the false positive rate—also known as false alarm rate—, namely, the probability that a negative sample (i.e., a non-keyword in our case) is wrongly classified as a positive one [217]:

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}. \tag{16}$$

Then, a better and prominent way of evaluating the performance of a KWS system is by means of the receiver operating characteristic (ROC) curve, which consists of the plot of pairs of false positive and true positive rate values that are obtained by sweeping the sensitivity (decision) threshold [218]. The left part of Figure 12 outlines example

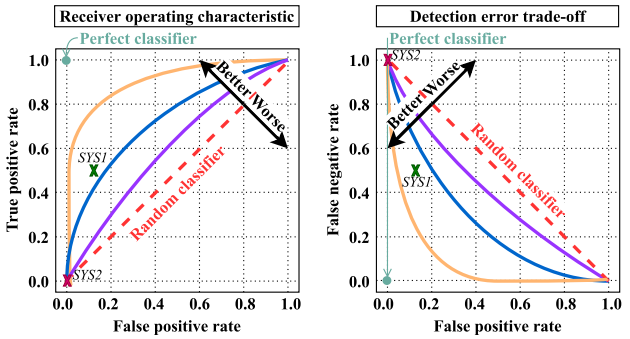

FIGURE 12. Outlining of the receiver operating characteristic (left) and detection error trade-off (right) curves. The location of *SYS1* and *SYS2* is indicated by green and red crosses, respectively. See the text for further explanation.

ROC curves. Coordinate (FPR $=0$, TPR $=1$) in the upper left corner represents a perfect classifier. The closer to this point a ROC curve is, the better a classification system. In addition, a system performing on the ROC space identity line would be randomly guessing. The area under the curve (AUC), which equals the probability that a classifier ranks a randomly-chosen positive sample higher than a randomly-chosen negative one [218], is also often employed as a ROC summary for KWS evaluation, e.g., [76], [85], [123], [145], [152], [219]–[221]. The larger the AUC $\in [0, 1]$, the better a system is [222].

Let us return for a moment to the example of Figure 11. It is easy to check that the KWS systems *SYS1* and *SYS2* would be characterized, in the ROC space, by the coordinates (FPR $=0.125$, TPR $=0.5$) and (FPR $=0$, TPR $=0$), respectively (see Figure 12). Unlike what happened when using accuracy, now we can rightly assess that *SYS1* (above the random guessing line) is much better than *SYS2* (on the random guessing line).

An alternative (with no particular preference) to the ROC curve (e.g., [24], [138], [177], [223]) is the detection error trade-off (DET) curve [224]. From the right part of Figure 12, it can be seen that a DET curve is like a ROC curve except for the y-axis being false negative rate —also known as miss rate [225]—, FNR:

$$\text{FNR} = \frac{\text{FN}}{\text{FN} + \text{TP}}. \tag{17}$$

This time, coordinate (FPR $=0$, FNR $=0$) in the bottom left corner represents a perfect classifier. The closer to this point a DET curve is, the better a classification system. Therefore, the smaller the AUC $\in [0, 1]$ in this case, the better a system is. Notice that, as FNR $=1-$ TPR, the DET curve is nothing else but a vertically-flipped version of the ROC curve. From the DET curve we can also straightforwardly obtain the equal error rate (EER) as the intersection point between the identity line and the DET curve (i.e., the point at which FNR $=$ FPR) [226]. Certainly, the lower the EER value, the better. Though the use of EER is much more widespread in the field of speaker verification [227]–[229], this DET summary

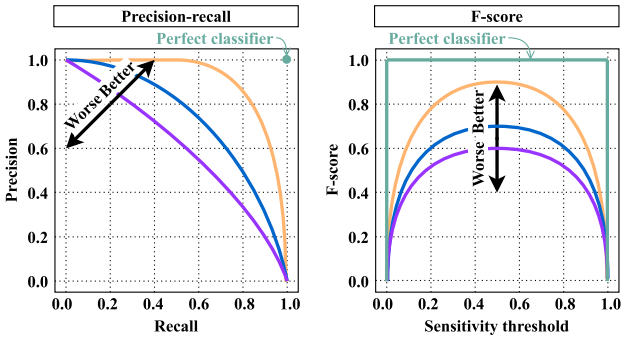

FIGURE 13. Outlining of the precision-recall (left) and F-score (right) curves. See the text for further explanation.

is sometimes considered for KWS evaluation [4], [76], [117], [123], [159], [220], [230].

In real-world KWS applications, typically, the cost of a false alarm is significantly greater than that of a miss detection[17] [231]. This is for example the case for voice activation of voice assistants, where privacy is a major concern [232] since this application involves streaming voice to a cloud server. As a result, a popular variant of the ROC and DET curves is that one replacing false positive rate along the x-axis by the number of false alarms per hour [8], [28], [31], [59], [60], [156], [162], [200]. By this, a practitioner can just set a very small number of false alarms per hour (e.g., 1) and identify the system with the highest (lowest) true positive (false negative) rate for deployment. An alternative good selection criterion consists of picking up the system maximizing, at a particular system-dependent sensitivity threshold, the so-called term-weighted value (TWV) [88], [231], [233]–[238]. Given a sensitivity threshold, TWV is a weighted linear combination of the false negative and false positive rates as in

$$\text{TWV} = 1 - (\text{FNR} + \beta \text{FPR}), \tag{18}$$

where $\beta \gg 1$ (e.g., $\beta = 999.9$ [231]) is a constant expressing the greater cost of a false alarm with respect to that of a miss detection.

C. PRECISION-RECALL AND F-SCORE CURVES

The precision-recall curve [239] is another important visual performance analysis tool for KWS systems (e.g., [12], [77], [129], [140]). Let precision, also known as positive predictive value [240], be the probability that a sample that is classified as positive is actually a positive sample:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{19}$$

Then, the precision-recall curve plots pairs of recall (equivalently, TPR, see Eq. (15)) and precision values that, as in the case of the ROC and DET curves, are obtained by sweeping the sensitivity threshold. This definition is schematized by the left part of Figure 13, where a perfect classifier lies on the coordinate (Recall $= 1$, Precision $= 1$). The closer to this point a precision-recall curve is and the larger the AUC $\in [0, 1]$, the better a classifier. This time, a (precision-recall) random guessing line has not been drawn, since it depends on the proportion of the positive class within both classes [240]. For example, while in a balanced scenario random guessing would be characterized by a horizontal line at a precision of 0.5, we can expect that such a line is closer to 0 precision in the event of the KWS problem due to the highly imbalance nature of it.

The close relationship between the ROC (and DET) and precision-recall curves can be intuited, and, in fact, there exists a one-to-one correspondence between both of them [239]. However, the precision-recall curve is considered to be a more informative visual analysis tool than the ROC one in our context [240]. This is because, thanks to the use of precision, the precision-recall curve allows us to better focus on the *minority* positive (i.e., keyword) class of interest (see Eq. (19)). On the precision-recall plane, while *SYS1* lies on the point (Recall $= 0.5$, Precision $= 0.5$), precision is undefined (i.e., Precision $= 0/0$) for *SYS2*, which should alert us to the existence of a problem with the latter system.

From precision and recall we can formulate the F-score metric [241], F_1, which is often used for KWS evaluation, e.g., [12], [129], [130], [140], [151], [242]. F-score is the harmonic mean of precision and recall, that is,

$$F_1 = \frac{2}{\text{Recall}^{-1} + \text{Precision}^{-1}} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}, \tag{20}$$

where $0 \leq F_1 \leq 1$, and the larger F_1, the better. Indeed, as for precision and recall, F-score can be calculated as a function of the sensitivity threshold and plotted as exemplified by the right part of Figure 13. In this representation, we assume that a KWS system provides confidence scores resulting from posterior probabilities, and this is why the sensitivity threshold ranges from 0 to 1. The larger the AUC $\in [0, 1]$, the better a system is. A perfect classifier would be characterized by an AUC of 1. As in the case of the precision-recall curve, a random guessing line has not been drawn either on the F-score space, since this similarly depends on the proportion between the positive and negative classes. Finally, let us notice that F-score is 0.5 and 0 for *SYS1* and *SYS2*, respectively, which clearly indicates the superiority of *SYS1* with respect to *SYS2*.

X. PERFORMANCE COMPARISON

In this section, we present a performance comparison among some of the latest and most relevant deep KWS systems reviewed throughout this manuscript. This comparison is carried out in terms of both KWS performance and computational complexity of the acoustic model, which is the main distinctive component of every system.

To measure KWS performance, we examine accuracy of systems in non-streaming mode on the Google Speech Commands Dataset (GSCD) *v1* and *v2* (described in Subsection VIII-A), which standardize 10 keywords (see Table 2).

[17] Evidently, in these circumstances, EER may not be a good metric candidate for system comparison.

TABLE 3. Performance comparison among some of the latest deep KWS systems in terms of both accuracy (%) and computational complexity (i.e., number of parameters and multiplications) of the acoustic model. Accuracy, provided with confidence intervals for some systems, is on the Google Speech Commands Dataset (GSCD) *v1* and *v2*. The reported values are directly taken from the references in the "Description" column. Unknown information is indicated by hyphens.

ID	Description	Year	Accuracy (%)		Computational complexity	
			GSCD *v1*	GSCD *v2*	No. of params.	No. of mults.
1	Standard FFNN with a pooling layer [32]	2020	91.2	90.6	447k	–
2	DenseNet with trainable window function and mixup data augmentation [67]	2018	92.8	–	–	–
3	Two-stage TDNN [58]	2018	94.3	–	251k	25.1M
4	CNN with striding [32]	2018	95.4	95.6	529k	–
5	BiLSTM with attention [133]	2018	95.6	96.9	202k	–
6	Residual CNN `res15` [30]	2018	95.8 ± 0.484	–	238k	894M
7	TDNN with shared weight self-attention [16]	2019	95.81 ± 0.191	–	12k	403k
8	DenseNet+BiLSTM with attention [48]	2019	96.2	97.3	223k	–
9	Residual CNN with temporal convolutions `TC-ResNet14` [50]	2019	96.2	–	137k	–
10	SVDF [32]	2019	96.3	96.9	354k	–
11	SincConv+(Grouped DS-CNN) [70]	2020	96.4	97.3	62k	–
12	Graph convolutional network `CENet-40` [49]	2019	96.4	–	61k	16.18M
13	GRU [32]	2020	96.6	97.2	593k	–
14	SincConv+(DS-CNN) [70]	2020	96.6	97.4	122k	–
15	Temporal CNN with depthwise convolutions `TENet12` [52]	2020	96.6	–	100k	2.90M
16	Residual DS-CNN with squeeze-and-excitation `DS-ResNet18` [51]	2020	96.71 ± 0.195	–	72k	285M
17	`TC-ResNet14` with neural architecture search `NoisyDARTS-TC14` [146]	2021	96.79 ± 0.30	97.18 ± 0.26	108k	6.3M
18	LSTM [32]	2020	96.9	97.5	–	–
19	DS-CNN with striding [32]	2018	97.0	97.1	485k	–
20	CRNN [32]	2020	97.0	97.5	467k	–
21	BiGRU with multi-head attention [32]	2020	97.2	98.0	743k	–
22	CNN with neural architecture search `NAS2_6_36` [125]	2020	97.22	–	886k	–
23	Keyword Transformer `KWT-3` [90]	2021	97.49 ± 0.15	98.56 ± 0.07	5.3M	–
24	Variant of TC-ResNet with self-attention `LG-Net6` [91]	2021	97.67	96.79	313k	–
25	Broadcasted residual CNN `BC-ResNet-8` [100]	2021	98.0	98.7	321k	89.1M

In this way, since the publicly available GSCD has become the *de facto* open benchmark for deep KWS, we can straightforwardly use accuracy values reported in the literature in order to rank the most prominent deep KWS systems. Regarding accuracy as an evaluation metric, recall that this metric, although not ideal, is still meaningful under the GSCD experimental conditions to explain the goodness of KWS systems, as discussed in Subsection IX-A.

On the other hand, the number of parameters and multiplications of the acoustic model is used to evaluate the computational complexity of the systems. Notice that these measures are a good approximation to the complexity of the entire deep KWS system since the acoustic model is, by far, the most demanding component in terms of computation. Actually, in [86], Tang *et al.* show that the number of parameters and, especially, the number of multiplications of the acoustic model are solid proxies predicting the power consumption of these systems.

Table 3 shows a performance comparison among some of the latest deep KWS systems in terms of both accuracy on the GSCD *v1* and *v2* (in percentages), and complexity of the acoustic model. The reported values are directly taken from the references in the "Description" column, while hyphens indicate non-available information. Notice that some of the accuracy values in Table 3 are shown along with confidence intervals that are calculated across different acoustic models trained with different random model initialization. Deep KWS systems are listed in ascending order in terms of their accuracy on the first version of the GSCD. From Table 3, it can be observed that KWS performance on GSCD *v2* tends to be slightly better than that on the first version of this dataset. This behavior could be related to the fact that the second version of this dataset has more word samples (see Table 1), which might lead to better trained acoustic models.

Also from Table 3, we can see the wide variety of architectures (e.g., standard FFNNs, SVDFs, TDNNs, CNNs, RNNs and CRNNs) integrating different elements (e.g., attention, residual connections and/or depthwise separable convolutions) that has been explored for deep KWS. It is not surprising that the worst-performing system is that whose acoustic model is based on a standard and relatively heavy (447k parameters) FFNN [32] (ID 1 in Table 3). Besides, the most frequently used acoustic model type is based on CNN. This surely is because CNNs are able to provide a highly competitive performance —thanks to exploiting local speech time-frequency correlations— while typically involving lesser

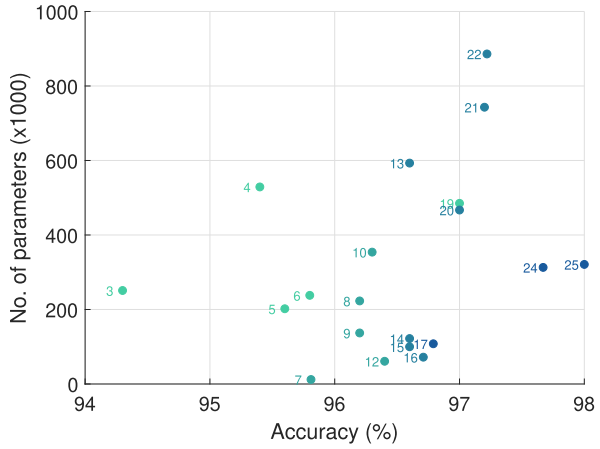

FIGURE 14. Location of some of the deep KWS systems of Table 3 on the plane defined by the dimensions "number of parameters" and "accuracy" (on the Google Speech Commands Dataset *v1*). Better systems can be found on the lower right corner of this plane. The systems are identified by the numbers in the "ID" column of Table 3. More recent systems are marked with a darker color.

computational complexity than other well-performing types of models like RNNs.

Furthermore, it is interesting to note the capability of neural architecture search techniques [243] to automatically produce acoustic models performing better than those manually designed. Thus, the performance of the residual CNN with temporal convolutions `TC-ResNet14` [50] (ID 9) is improved when `NoisyDARTS-TC14` [146] (ID 17) automatically searches for kernel sizes, additional skip connections and enabling or not squeeze-and-excitation [244]. Even better, this is achieved by employing fewer parameters, i.e., 137k *versus* 108k. In addition, the CNN with neural architecture search `NAS2_6_36` [125] (ID 22) reaches an outstanding performance (97.22% accuracy on the GSCD *v1*), though at the expense of using a large number of parameters (886k).

The effectiveness of CRNNs combining CNNs and RNNs (see Subsection IV-C) can also be assessed from Table 3. For instance, the combination of DenseNet[18] [131] with a BiLSTM network with attention as in [48] (ID 8) yields superior KWS accuracy with respect to considering standalone either DenseNet [67] (ID 2) or a BiLSTM network with attention [133] (ID 5). Moreover, we can see that the performance of a rather basic CRNN incorporating a GRU layer [32] (ID 20) is quite competitive.

Due to the vast number of disparate factors contributing to the performance of the deep KWS systems of Table 3, it is extremely difficult to draw strong conclusions and even trends far beyond the ones indicated above. Figure 14 gives another perspective of Table 3 by plotting the location of some of the systems of this table on the plane defined by the dimensions "number of parameters" and "accuracy" (on the GSCD *v1*). In this figure, the systems are identified by the numbers in the "ID" column of Table 3.

Since more recent deep KWS systems are marked with a darker color in Figure 14, it can be clearly observed that, primarily, the driving force is the optimization of KWS performance, where the computational complexity, although important, is relegated to a secondary position. A good example of this is the so-called Keyword Transformer `KWT-3` [90] (ID 23), a fully self-attentional Transformer [144] that is an adaptation of Vision Transformer [245] to the KWS task. `KWT-3` (not included in Figure 14), which achieves state-of-the-art performance (97.49% and 98.56% accuracy on the GSCD *v1* and *v2*, respectively), has the extraordinary amount of more than 5 million parameters. That being said, generally, we will be more interested in systems exhibiting both high accuracy and a small footprint, i.e., in systems that can be found on the lower right corner of the plane in Figure 14. In this region of the plane we have the following two groups of systems:

1) *Systems With IDs 14, 15, 16 and 17:* These systems are characterized by a good KWS performance along with a particularly reduced number of parameters. All of them are based on CNNs while most of them integrate residual connections and/or depthwise separable convolutions. Furthermore, the three best performing systems (with IDs 15, 16 and 17) integrate either dilated or temporal convolutions to exploit long time-frequency dependencies.
2) *Systems With IDs 24 and 25:* These two systems are characterized by an outstanding KWS performance along with a relatively small number of parameters. Both of them are based on CNNs and they integrate residual connections and a mechanism to exploit long time-frequency dependencies: dilated convolutions in System 25, and temporal convolutions and self-attention layers in System 24. System 25 also incorporates depthwise separable convolutions.

The analysis of the above two groups of systems very much reinforces our summary reflections concluding Subsection IV-B. In other words, a state-of-the-art KWS system comprising a CNN-based acoustic model should cover the following three elements in order to reach a high performance with a small footprint: a mechanism to exploit long time-frequency dependencies, depthwise separable convolutions [136] and residual connections [126].

XI. AUDIO-VISUAL KEYWORD SPOTTING

In face-to-face human communication, observable articulators like the lips are an important information source. In other words, human speech perception is bimodal, since it relies on both auditory and visual information. Similarly, speech processing systems such as ASR systems can be benefited from exploiting visual information along with the audio information to enhance their performance [246]–[249]. This can be particularly fruitful in real-world scenarios where severe acoustic distortions (e.g., strong background noise and

[18]Recall that DenseNet is an extreme case of residual CNN with a hive of skip connections.

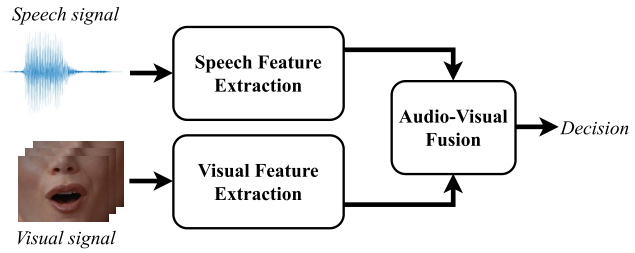

FIGURE 15. General diagram of a modern audio-visual keyword spotting system.

reverberation) are present, since the visual information is not affected by acoustic distortions.

While fusion of audio-visual information is a quite active research area in ASR (e.g., see [246]–[249]), very few works have studied it for (deep) KWS [250]–[252]. This means that audio-visual KWS is a new and essentially unexplored field. Nevertheless, we believe that this emerging area could become extremely important in future applications, and this is why we devote the present section to audio-visual KWS.

Figure 15 illustrates the general diagram of a modern audio-visual KWS system. First, speech and visual features are extracted. In former audio-visual KWS work [250], [251], visual feature extraction consists of a pipeline comprising face detection and lip localization (via landmark estimation), and visual feature extraction itself from the lips crop. Nowadays, the use of a deep learning model fed with raw images containing the uncropped speaker's face seems to be the preferred approach for visual feature extraction [252]. Finally, the extracted audio-visual information is fused in order to come up with a decision about the presence or not of a keyword. Typically, one of the two following fusion strategies is considered in practice [253]:

1) *Feature-Level Fusion:* Speech and visual features are somehow combined (e.g., concatenated) before their joint classification using a neural network model.
2) *Decision-Level Fusion:* The final decision is formed from the combination of the decisions from separate speech and visual neural network-based classifiers. This well-performing approach seems to be preferred [250]–[252] over the feature-level fusion scheme and is less data-hungry than feature-level fusion [250].

Notice that thanks to the integration of visual information —which, as aforementioned, is not affected by acoustic distortions—, audio-visual KWS achieves the greatest relative improvements with respect to audio-only KWS at lower SNRs [250]–[252].

For those who are interested in audio-visual KWS research, the following realistic and challenging audio-visual benchmarks can be of interest: Lip Reading in the Wild (LRW) [254], and Lip Reading Sentences 2 (LRS2) [255] and 3 (LRS3) [256] datasets. While LRW comprises single-word utterances from BBC TV broadcasts, LRS2 and LRS3 consist of thousands of spoken sentences from BBC TV and TED(x) talks, respectively.

XII. CONCLUSION AND FUTURE DIRECTIONS

The goal of this article has been to provide a comprehensive overview of state-of-the-art KWS technology, namely, of deep KWS. We have seen that the core of this paradigm is a DNN-based acoustic model whose goal is the generation, from speech features, of posterior probabilities that are subsequently processed to detect the presence of a keyword. Deep spoken KWS has revitalized KWS research by enabling a massive deployment of this technology for real-world applications, especially in the area of voice assistant activation.

We foresee that, as has been happening to date, advances in ASR research will dramatically continue impacting the field of KWS. In particular, we think that the expected progress in end-to-end ASR [257] (replacing handcrafted speech features by optimal feature learning integrated in the acoustic model) will also be reflected in KWS.

Immediate future work will keep focusing on advancing acoustic modeling towards two goals simultaneously: *1)* improving KWS performance in real-life acoustic conditions, and *2)* computational complexity reduction. With these two goals in mind, surely, acoustic model research will be mainly focused on the development of novel and efficient convolutional blocks. This is because of the good properties of CNNs allowing us to achieve an outstanding performance with a small footprint, as has been widely discussed throughout this paper. Furthermore, based on its promising initial results for KWS [125], [146], we expect that neural architecture search will play a greater role in acoustic model architecture design.

Specifically within the context of computational complexity reduction, acoustic model compression will be, more than ever, a salient research line [101]. Indeed, this is driven by the numerous applications of KWS that involve embedding KWS technology in small electronic devices characterized by severe memory, computation and power constraints. Acoustic model compression entails three major advantages: *1)* reduced memory footprint, *2)* decreased inference latency, and *3)* less energy consumption. All of this is of utmost importance for, e.g., enabling on-device acoustic model re-training for robustness purposes or personalized keyword inclusion. Acoustic model compression research will undoubtedly encompass model parameter quantization, neural network pruning and knowledge distillation [258], among other approaches.

Another line of research that might experience a notable growth in the short term could be semi-supervised learning [259] for KWS. Especially in an industrial environment, it is simple to daily collect a vast amount of speech data from users of cloud speech services. These data are potentially valuable to strengthen KWS acoustic models. However, the cost of labeling such an enormous amount of data for discriminative model training can easily be prohibitively expensive. To not "waste" these unlabeled speech data, semi-supervised

learning methodologies can help by allowing hybrid learning based on both small and big volumes of labeled and unlabeled data, respectively.

On the other hand, consumers seem to increasingly demand or, at least, value a certain degree of personalization when it comes to consumer electronics. While some research has already addressed some KWS personalization aspects (as we have discussed in this article), we foresee that KWS personalization will become even more relevant in the immediate future. This means that we can expect new research going deeper into topics like *efficient* open-vocabulary (personalized) KWS and joint KWS and speaker verification [260], [261].

Last but not least, recall that KWS technology is many times intended to run on small devices like smart speakers and wearables that typically embed more than one microphone. This type of multi-channel information has been successfully leveraged by ASR in different ways (which includes, e.g., beamforming) to provide robustness against acoustic distortions [44], [262]. Surprisingly, and as previously outlined in Section VI, multi-channel KWS has only been marginally studied. Therefore, we expect that this rather unexplored area is worthy to be examined, which could lead to contributions further improving KWS performance in real-life (i.e., noisy) conditions.

REFERENCES

[1] M. B. Hoy, "Alexa, Siri, Cortana, and more: An introduction to voice assistants," *Med. Reference Services Quart.*, vol. 37, no. 1, pp. 81–88, 2018.

[2] A. H. Michaely, X. Zhang, G. Simko, C. Parada, and P. Aleksic, "Keyword spotting for Google Assistant using contextual speech recognition," in *Proc. IEEE Autom. Speech Recognit. Understand. Workshop*, Okinawa, Japan, Oct. 2017, pp. 272–278.

[3] O. Vinyals and S. Wegmann, "Chasing the metric: Smoothing learning algorithms for keyword detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Florence, Italy, May 2014, pp. 3301–3305.

[4] Y. Zhuang, X. Chang, Y. Qian, and K. Yu, "Unrestricted vocabulary keyword spotting using LSTM-CTC," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, San Francisco, CA, USA, Sep. 2016, pp. 938–942.

[5] M. Weintraub, "Keyword-spotting using SRI's DECIPHER large-vocabulary speech-recognition system," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Minneapolis, MI, USA, Apr. 1993, pp. 463–466.

[6] D. R. H. Miller, M. Kleber, C.-L. Kao, O. Kimball, T. Colthurst, S. A. Lowe, R. M. Schwartz, and H. Gish, "Rapid and accurate spoken term detection," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Antwerp, Belgium, Aug. 2007, pp. 314–317.

[7] G. Chen, O. Yilmaz, J. Trmal, D. Povey, and S. Khudanpur, "Using proxies for OOV keywords in the keyword search task," in *Proc. IEEE Autom. Speech Recognit. Understand. Workshop*, Olomouc, Czech Republic, Dec. 2013, pp. 416–421.

[8] Y. Wang and Y. Long, "Keyword spotting based on CTC and RNN for Mandarin Chinese speech," in *Proc. Int. Symp. Chin. Spoken Lang. Process.*, Taipei, Taiwan, Nov. 2018, pp. 374–378.

[9] C. Shan, J. Zhang, Y. Wang, and L. Xie, "Attention-based end-to-end models for small-footprint keyword spotting," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Hyderabad, India, Sep. 2018, pp. 2037–2041.

[10] R. Rikhye, Q. Wang, Q. Liang, Y. He, D. Zhao, Y. A. Huang, A. Narayanan, and I. McGraw, "Personalized keyphrase detection using speaker and environment information," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Brno, Czechia, Sep./Oct. 2021, pp. 4204–4208.

[11] G. Chen, C. Parada, and T. N. Sainath, "Query-by-example keyword spotting using long short-term memory networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Brisbane, MD, Australia, Feb. 2015, pp. 5236–5240.

[12] S. Chai, Z. Yang, C. Lv, and W.-Q. Zhang, "An end-to-end model based on TDNN-BiGRU for keyword spotting," in *Proc. Int. Conf. Asian Lang. Process.*, Shanghai, China, Nov. 2019, pp. 402–406.

[13] M. Sun, D. Snyder, Y. Gao, V. Nagaraja, M. Rodehorst, S. Panchapagesan, N. Strom, S. Matsoukas, and S. Vitaladevuni, "Compressed time delay neural network for small-footprint keyword spotting," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Stockholm, Sweden, Aug. 2017, pp. 3607–3611.

[14] D. Can and M. Saraclar, "Lattice indexing for spoken term detection," *IEEE Trans. Audio, Speech, Language Process.*, vol. 19, pp. 2338–2347, 2011.

[15] R. Prabhavalkar, R. Alvarez, C. Parada, P. Nakkiran, and T. N. Sainath, "Automatic gain control and multi-style training for robust small-footprint keyword spotting with deep neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Brisbane, MD, Australia, May 2015, pp. 4704–4708.

[16] Y. Bai, J. Yi, J. Tao, Z. Wen, Z. Tian, C. Zhao, and C. Fan, "A time delay neural network with shared weight self-attention for small-footprint keyword spotting," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Graz, Austria, Sep. 2019, pp. 2190–2194.

[17] J. Hou, Y. Shi, M. Ostendorf, M.-Y. Hwang, and L. Xie, "Region proposal network based small-footprint keyword spotting," *IEEE Signal Process. Lett.*, vol. 26, no. 10, pp. 1471–1475, Oct. 2019.

[18] J. Rohlicek, W. Russell, S. Roukos, and H. Gish, "Continuous hidden Markov modeling for speaker-independent word spotting," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Glasgow, U.K., May 1989, pp. 627–630.

[19] R. Rose and D. Paul, "A hidden Markov model based keyword recognition system," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Albuquerque, NM, USA, Apr. 1990, pp. 129–132.

[20] J. Wilpon, L. Miller, and P. Modi, "Improvements and applications for key word recognition using hidden Markov modeling techniques," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Toronto, ON, Canada, Apr. 1991, pp. 309–312.

[21] I.-F. Chen and C.-H. Lee, "A hybrid HMM/DNN approach to keyword spotting of short words," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Lyon, France, Apr. 2013, pp. 1574–1578.

[22] G. Chen, C. Parada, and G. Heigold, "Small-footprint keyword spotting using deep neural networks," in *Proc. IEEE Int. Conf. Acoustics, Speech Signal Process.*, Florence, Italy, May 2014, pp. 4087–4091.

[23] M. Sun, V. Nagaraja, B. Hoffmeister, and S. Vitaladevuni, "Model shrinking for embedded keyword spotting," in *Proc. IEEE Int. Conf. Mach. Learn. Appl.*, Miami, FL, USA, Dec. 2015, pp. 369–374.

[24] S. Panchapagesan, M. Sun, A. Khare, S. Matsoukas, A. Mandal, B. Hoffmeister, and S. Vitaladevuni, "Multi-task learning and weighted cross-entropy for DNN-based keyword spotting," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, San Francisco, CA, USA, Sep. 2016, pp. 760–764.

[25] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 2, pp. 260–269, Apr. 1967.

[26] X. Wang, S. Sun, C. Shan, J. Hou, L. Xie, S. Li, and X. Lei, "Adversarial examples for improving end-to-end attention-based small-footprint keyword spotting," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Brighton, U.K., May 2019, pp. 6366–6370.

[27] B. D. Scott and M. E. Rafn, "Suspending noise cancellation using keyword spotting," U.S. Patent 9 398 367, Jul. 19, 2016. [Online]. Available: https://www.google.com/patents/US9398367B1

[28] T. N. Sainath and C. Parada, "Convolutional neural networks for small-footprint keyword spotting," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Dresden, Germany, Sep. 2015, pp. 1478–1482.

[29] M. Sun, A. Raju, G. Tucker, S. Panchapagesan, G. Fu, A. Mandal, S. Matsoukas, N. Strom, and S. Vitaladevuni, "Max-pooling loss training of long short-term memory networks for small-footprint keyword spotting," in *Proc. IEEE Spoken Lang. Technol. Workshop*, San Diego, CA, USA, Dec. 2016, pp. 474–480.

[30] R. Tang and J. Lin, "Deep residual learning for small-footprint keyword spotting," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Calgary, AB, Canada, Apr. 2020, pp. 5484–5488.

[31] R. Alvarez and H.-J. Park, "End-to-end streaming keyword spotting," in *Proc. IEEE Int. Conf. Acoustics, Speech Signal Process.*, Brighton, U.K., May 2019, pp. 6336–6340.

[32] O. Rybakov, N. Kononenko, N. Subrahmanya, M. Visontai, and S. Laurenzo, "Streaming keyword spotting on mobile devices," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Shanghai, China, Oct. 2020, pp. 2277–2281.

[33] B. K. Deka and P. Das, "A review of keyword spotting as an audio mining technique," *Int. J. Comput. Sci. Eng.*, vol. 7, no. 1, pp. 757–769, Jan. 2019.

[34] S. Tabibian, "A survey on structured discriminative spoken keyword spotting," *Artif. Intell. Rev.*, vol. 53, pp. 2483–2520, Oct. 2020.

[35] L. Mary, *Speech Databases: Features, Techniques and Evaluation Measures*, A. Neustein, Ed. Cham, Switzerland: Springer, 2018.

[36] A. Mandal, K. R. Prasanna Kumar, and P. Mitra, "Recent developments in spoken term detection: A survey," *Int. J. Speech Technol.*, vol. 17, no. 2, pp. 183–198, Jun. 2014.

[37] D. Yu and J. Li, "Recent progresses in deep learning based acoustic models," *IEEE/CAA J. Autom. Sinica*, vol. 4, no. 3, pp. 396–409, Jul. 2017.

[38] D. Wang, X. Wang, and S. Lv, "An overview of end-to-end automatic speech recognition," *MDPI Symmetry*, vol. 11, pp. 1–27, 2019.

[39] J. Hou, Y. Shi, M. Ostendorf, M.-Y. Hwang, and L. Xie, "Mining effective negative training samples for keyword spotting," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Barcelona, Spain, May 2020, pp. 7444–7448.

[40] J. Wang, Y. He, C. Zhao, Q. Shao, W.-W. Tu, T. Ko, H. Y. Lee, and L. Xie, "Auto-KWS 2021 challenge: Task, datasets, and baselines," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Brno, Czechia, Aug. 2021, pp. 4244–4248.

[41] B. U. Pedroni, S. Sheik, H. Mostafa, S. Paul, C. Augustine, and G. Cauwenberghs, "Small-footprint spiking neural networks for power-efficient keyword spotting," in *Proc. IEEE Biomed. Circuits Syst. Conf.*, Cleveland, OH, USA, Oct. 2018, pp. 1–4.

[42] B. Liu, S. Nie, Y. Zhang, S. Liang, Z. Yang, and W. Liu, "Focal loss and double-edge-triggered detector for robust small-footprint keyword spotting," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Brighton, U.K., May 2019, pp. 6361–6365.

[43] P. M. Sørensen, B. Epp, and T. May, "A depthwise separable convolutional neural network for keyword spotting on an embedded system," *EURASIP J. Audio, Speech, Music Process.*, vol. 2020, no. 1, pp. 1–14, Dec. 2020.

[44] I. López-Espejo, "Robust speech recognition on intelligent mobile devices with dual-microphone," Ph.D. dissertation, Dept. Signal Theory, Telematics Commun., Univ. Granada, Granada, Spain, 2017.

[45] X. Wang, S. Sun, and L. Xie, "Virtual adversarial training for DS-CNN based small-footprint keyword spotting," in *Proc. IEEE Autom. Speech Recognit. Understand. Workshop*, Singapore, Dec. 2019, pp. 607–612.

[46] S. Fernández, A. Graves, and J. Schmidhuber, "An application of recurrent neural networks to discriminative keyword spotting," in *Proc. Int. Conf. Artif. Neural Netw.*, Porto, Portugal, Sep. 2007, pp. 220–229.

[47] E. A. Ibrahim, J. Huisken, H. Fatemi, and J. P. de Gyvez, "Keyword spotting using time-domain features in a temporal convolutional network," in *Proc. Euromicro Conf. Digit. Syst. Design*, Kallithea, Greece, Aug. 2019, pp. 313–319.

[48] M. Zeng and N. Xiao, "Effective combination of DenseNet and BiLSTM for keyword spotting," *IEEE Access*, vol. 7, pp. 10767–10775, 2019.

[49] X. Chen, S. Yin, D. Song, P. Ouyang, L. Liu, and S. Wei, "Small-footprint keyword spotting with graph convolutional network," in *Proc. Autom. Speech Recognit. Understand. Workshop*, Singapore, Dec. 2019, pp. 539–546.

[50] S. Choi, S. Seo, B. Shin, H. Byun, M. Kersner, B. Kim, D. Kim, and S. Ha, "Temporal convolution for real-time keyword spotting on mobile devices," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Graz, Austria, Sep. 2019, pp. 3372–3376.

[51] M. Xu and X.-L. Zhang, "Depthwise separable convolutional ResNet with squeeze-and-excitation blocks for small-footprint keyword spotting," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Shanghai, China, Oct. 2020, pp. 2547–2551.

[52] X. Li, X. Wei, and X. Qin, "Small-footprint keyword spotting with multi-scale temporal convolution," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Shanghai, China, Oct. 2020, pp. 1987–1991.

[53] E. Yılmaz, J. Wu, Y. Chen, X. Meng, and H. Li, "Deep convolutional spiking neural networks for keyword spotting," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Shanghai, China, Oct. 2020, pp. 2557–2561.

[54] Z.-H. Tan, A. Sarkar, and N. Dehak, "RVAD: An unsupervised segment-based robust voice activity detection method," *Comput. Speech Lang.*, vol. 59, pp. 1–21, Jan. 2020.

[55] L. Lugosch and S. Myer, "DONUT: CTC-based query-by-example keyword spotting," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2018, pp. 1–9.

[56] Y. Yuan, Z. Lv, S. Huang, and L. Xie, "Verifying deep keyword spotting detection with acoustic word embeddings," in *Proc. IEEE Autom. Speech Recognit. Understand. Workshop*, Singapore, Dec. 2019, pp. 613–620.

[57] C. Yang, X. Wen, and L. Song, "Multi-scale convolution for robust keyword spotting," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Shanghai, China, Oct. 2020, pp. 2577–2581.

[58] S. Myer and V. S. Tomar, "Efficient keyword spotting using time delay neural networks," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Hyderabad, India, Sep. 2018, pp. 1264–1268.

[59] T. Higuchi, M. Ghasemzadeh, K. You, and C. Dhir, "Stacked 1D convolutional networks for end-to-end small footprint voice trigger detection," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Shanghai, China, Oct. 2020, pp. 2592–2596.

[60] Y. He, R. Prabhavalkar, K. Rao, W. Li, A. Bakhtin, and I. McGraw, "Streaming small-footprint keyword spotting using sequence-to-sequence models," in *Proc. Automatic Speech Recognit. Understand. Workshop*, Okinawa, Japan, Dec. 2017, pp. 474–481.

[61] X. Xuan, M. Wang, X. Zhang, and F. Sun, "Robust small-footprint keyword spotting using sequence-to-sequence model with connectionist temporal classifier," in *Proc. Int. Conf. Inf. Commun. Signal Process.*, Weihai, China, Sep. 2019, pp. 400–404.

[62] E. Sharma, G. Ye, W. Wei, R. Zhao, Y. Tian, J. Wu, L. He, E. Lin, and Y. Gong, "Adaptation of RNN transducer with text-to-speech technology for keyword spotting," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Barcelona, Spain, May 2020, pp. 7484–7488.

[63] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, Pittsburgh, PA, USA, Jun. 2006, pp. 369–376.

[64] A. Graves, "Sequence transduction with recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, Edinburgh, Scotland, Jul. 2012, pp. 1–19.

[65] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Vancouver, BC, Canada, May 2013, pp. 6645–6649.

[66] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[67] X. Du, M. Zhu, M. Chai, and X. Shi, "End to end model for keyword spotting with trainable window function and Densenet," in *Proc. IEEE Int. Conf. Digit. Signal Process.*, Shanghai, China, Nov. 2018, pp. 1–5.

[68] M. Lee, J. Lee, H. J. Jang, B. Kim, W. Chang, and K. Hwang, "Orthogonality constrained multi-head attention for keyword spotting," in *Proc. IEEE Autom. Speech Recognit. Understand. Workshop*, Singapore, Dec. 2019, pp. 86–92.

[69] A. Riviello and J.-P. David, "Binary speech features for keyword spotting tasks," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Graz, Austria, Sep. 2019, pp. 3460–3464.

[70] S. Mittermaier, L. Kürzinger, B. Waschneck, and G. Rigoll, "Small-footprint keyword spotting on raw audio data with Sinc-convolutions," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Barcelona, Spain, May 2020, pp. 7454–7458.

[71] H.-J. Park, P. Violette, and N. Subrahmanya, "Learning to detect keyword parts and whole by smoothed max pooling," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Barcelona, Spain, Mar. 2020, pp. 7899–7903.

[72] H. Wu, Y. Jia, Y. Nie, and M. Li, "Domain aware training for far-field small-footprint keyword spotting," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Shanghai, China, Oct. 2020, pp. 2562–2566.

[73] J. S. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," *Neurocomputing*, vol. 4, pp. 227–236, May 1990.

[74] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[75] Y. Zhang, N. Suda, L. Lai, and V. Chandra, "Hello edge: Keyword spotting on microcontrollers," 2018, *arXiv:1711.07128v3*.

[76] R. Kumar, V. Yeruva, and S. Ganapathy, "On convolutional LSTM modeling for joint wake-word detection and text dependent speaker verification," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Hyderabad, India, Sep. 2018, pp. 1121–1125.

[77] Y. Tan, K. Zheng, and L. Lei, "An in-vehicle keyword spotting system with multi-source fusion for vehicle applications," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Marrakesh, Morocco, Sep. 2019, pp. 1–6.

[78] A. Coucke, M. Chlieh, T. Gisselbrecht, D. Leroy, M. Poumeyrol, and T. Lavril, "Efficient keyword spotting using dilated convolutions and gating," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Brighton, U.K., May 2019, pp. 6351–6355.
[79] Y. Gao, N. D. Stein, C.-C. Kao, Y. Cai, M. Sun, T. Zhang, and S. Vitaladevuni, "On front-end gain invariant modeling for wake word spotting," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Shanghai, China, Aug. 2020, pp. 991–995.
[80] S. S. Stevens, J. Volkmann, and E. B. Newman, "A Scale for the measurement of the psychological magnitude pitch," *J. Acoust. Soc. Amer.*, vol. 8, no. 3, pp. 185–190, Jan. 1937.
[81] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-28, no. 4, pp. 357–366, Aug. 1980.
[82] I. López-Espejo, Z.-H. Tan, and J. Jensen, "Exploring filterbank learning for keyword spotting," in *Proc. Eur. Signal Process. Conf.*, Amsterdam, The Netherlands, Jan. 2021, pp. 331–335.
[83] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural Networks: Tricks Trade*, vol. 7700. Berlin, Germany: Springer, 2012, pp. 9–48.
[84] M. Wöllmer, B. Schuller, and G. Rigoll, "Keyword spotting exploiting long short-term memory," *Speech Commun.*, vol. 55, pp. 252–265, Oct. 2013.
[85] T. Fuchs and J. Keshet, "Spoken term detection automatically adjusted for a given threshold," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 8, pp. 1310–1317, Dec. 2017.
[86] R. Tang, W. Wang, Z. Tu, and J. Lin, "An experimental analysis of the power consumption of convolutional neural networks for keyword spotting," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Calgary, AB, Canada, 2018, pp. 5479–5483.
[87] M. Muhsinzoda, C. C. Corona, D. A. Pelta, and J. L. Verdegay, "Activating accessible pedestrian signals by voice using keyword spotting systems," in *Proc. Int. Smart Cities Conf.*, Casablanca, Morocco, Oct. 2019, pp. 531–534.
[88] B. Pattanayak, J. K. Rout, and G. Pradhan, "Adaptive spectral smoothening for development of robust keyword spotting system," *IET Signal Process.*, vol. 13, no. 5, pp. 544–550, Jul. 2019.
[89] Y. Chen, T. Ko, L. Shang, X. Chen, X. Jiang, and Q. Li, "An investigation of few-shot learning in spoken term classification," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Shanghai, China, Oct. 2020, pp. 2582–2586.
[90] A. Berg, M. O'Connor, and M. T. Cruz, "Keyword transformer: A self-attention model for keyword spotting," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Brno, Czechia, Aug. 2021, pp. 4249–4253.
[91] L. Wang, R. Gu, N. Chen, and Y. Zou, "Text anchor based metric learning for small-footprint keyword spotting," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Brno, Czechia, Aug./Sep. 2021, pp. 4219–4223.
[92] S. Watanabe, M. Delcroix, F. Metze, and J. R. Hershey, *New Era for Robust Speech Recognition*. Cham, Switzerland: Springer, 2017.
[93] S. O. Arik, M. Kliegl, R. Child, J. Hestness, A. Gibiansky, C. Fougner, R. Prenger, and A. Coates, "Convolutional recurrent neural networks for small-footprint keyword spotting," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Stockholm, Sweden, Aug. 2017, pp. 1606–1610.
[94] Y. A. Huang, T. Z. Shabestary, and A. Gruenstein, "Hotword Cleaner: Dual-microphone adaptive noise cancellation with deferred filter coefficients for robust keyword spotting," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Brighton, U.K., May 2019, pp. 6346–6350.
[95] H. Mazzawi, X. Gonzalvo, A. Kracun, P. Sridhar, N. Subrahmanya, I. L. Moreno, H. J. Park, and P. Violette, "Improving keyword spotting and language identification via neural architecture search at scale," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Graz, Austria, Sep. 2019, pp. 1278–1282.
[96] M. Yu, X. Ji, B. Wu, D. Su, and D. Yu, "End-to-end multi-look keyword spotting," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Shanghai, China, Aug. 2020, pp. 66–70.
[97] X. Ji, M. Yu, J. Chen, J. Zheng, D. Su, and D. Yu, "Integration of multi-look beamformers for multi-channel keyword spotting," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Barcelona, Spain, May 2020, pp. 7464–7468.
[98] H. Yan, Q. He, and W. Xie, "CRNN-CTC based Mandarin keywords spotting," in *Proc. Int. Conf. Acoust., Speech Signal Process.*, Barcelona, Spain, May 2020, pp. 7489–7493.
[99] P. Zhang and X. Zhang, "Deep template matching for small-footprint and configurable keyword spotting," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Shanghai, China, Oct. 2020, pp. 2572–2576.
[100] B. Kim, S. Chang, J. Lee, and D. Sung, "Broadcasted residual learning for efficient keyword spotting," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Brno, Czechia, Aug. 2021, pp. 4538–4542.
[101] Y. Tian, H. Yao, M. Cai, Y. Liu, and Z. Ma, "Improving RNN transducer modeling for small-footprint keyword spotting," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Toronto, ON, Canada, Oct. 2021, pp. 5624–5628.
[102] J. Hou, L. Xie, and Z. Fu, "Investigating neural network based query-by-example keyword spotting approach for personalized wake-up word detection in Mandarin Chinese," in *Proc. Int. Symp. Chin. Spoken Lang. Process.*, Tianjin, China, Oct. 2016, pp. 1–5.
[103] N. Sacchi, A. Nanchen, M. Jaggi, and M. Cernak, "Open-vocabulary keyword spotting with audio and text embeddings," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Graz, Austria, Sep. 2019, pp. 3362–3366.
[104] J. Huang, W. Gharbieh, H. S. Shim, and E. Kim, "Query-by-example keyword spotting system using multi-head attention and soft-triple loss," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Toronto, ON, Canada, Jun. 2021, pp. 6858–6862.
[105] A. Singhal, "Modern information retrieval: A brief overview," *IEEE Comput. Soc. Tech. Committee Data Eng.*, vol. 24, no. 4, pp. 35–43, Jan. 2001.
[106] C. Parada, A. Sethy, and B. Ramabhadran, "Query-by-example spoken term detection for OOV terms," in *Proc. Autom. Speech Recognit. Understand. Workshop*, Moreno, Italy, Dec. 2009, pp. 404–409.
[107] K. Levin, K. Henry, A. Jansen, and K. Livescu, "Fixed-dimensional acoustic embeddings of variable-length segments in low-resource settings," in *Proc. IEEE Autom. Speech Recognit. Understand. Workshop*, Olomouc, Czech Republic, Dec. 2013, pp. 410–415.
[108] Y. Mishchenko, Y. Goren, M. Sun, C. Beauchene, S. Matsoukas, O. Rybakov, and S. N. P. Vitaladevuni, "Low-bit quantization and quantization-aware training for small-footprint keyword spotting," in *Proc. IEEE Int. Conf. Mach. Learn. Appl.*, Boca Raton, FL, USA, Dec. 2019, pp. 706–711.
[109] B. Liu, Y. Sun, and B. Liu, "Translational bit-by-bit multi-bit quantization for CRNN on keyword spotting," in *Proc. Int. Conf. Cyber-Enabled Distrib. Comput. Knowl. Discovery*, Guilin, China, Oct. 2019, pp. 444–451.
[110] 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 *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Calgary, AB, Canada, Aug. 2018, pp. 5349–5353.
[111] H. Muckenhirn, M.-Doss. Magimai, and S. Marcel, "Towards directly modeling raw speech signal for speaker verification using CNNs," in *Proc. Int. Conf. Acoust., Speech Signal Process.*, Calgary, AB, Canada, Apr. 2018, pp. 4884–4888.
[112] M. Ravanelli and Y. Bengio, "Speaker recognition from raw waveform with SincNet," in *Proc. IEEE Spoken Lang. Technol. Workshop*, Athens, Greece, Dec. 2018, pp. 1021–1028.
[113] T. Irino and M. Unoki, "An analysis/synthesis auditory filterbank based on an IIR implementation of the gammachirp," *J. Acoust. Soc. Jpn.*, vol. 20, pp. 397–406, 1999.
[114] T. N. Sainath, R. J. Weiss, A. Senior, K. W. Wilson, and O. Vinyals, "Learning the speech front-end with raw waveform CLDNNs," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Dresden, Germany, Dec. 2015, pp. 1–5.
[115] H. Seki, K. Yamamoto, and S. Nakagawa, "A deep neural network integrated with filterbank learning for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, New Orleans, LO, USA, Mar. 2017, pp. 5480–5484.
[116] N. Zeghidour, N. Usunier, G. Synnaeve, R. Collobert, and E. Dupoux, "End-to-end speech recognition from the raw waveform," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Hyderabad, India, Sep. 2018, pp. 781–785.
[117] R. Shankar, C. Vikram, and S. Prasanna, "Spoken keyword detection using joint DTW-CNN," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Hyderabad, India, Sep. 2018, pp. 117–121.
[118] E.-T. Albert, C. Lemnaru, M. Dinsoreanu, and R. Potolea, "Keyword spotting using dynamic time warping and convolutional recurrent networks," in *Proc. Int. Conf. Intell. Comput. Commun. Process.*, Cluj-Napoca, Romania, Nov. 2019, pp. 53–60.

[119] P. Nakkiran, R. Alvarez, R. Prabhavalkar, and C. Parada, "Compressing deep neural networks using a rank-constrained topology," in *Proc. Annus. Conf. Int. Speech Commun. Assoc.*, Dresden, Germany, Sep. 2015, pp. 1473–1477.

[120] H. Mostafa, "Supervised learning based on temporal coding in spiking neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 7, pp. 3227–3235, Jul. 2018.

[121] G. Tucker, M. Wu, M. Sun, S. Panchapagesan, G. Fu, and S. Vitaladevuni, "Model compression applied to small-footprint keyword spotting," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, San Francisco, CA, USA, Sep. 2016, pp. 1878–1882.

[122] Y. Huang, T. Hughes, T. Z. Shabestary, and T. Applebaum, "Supervised noise reduction for multichannel keyword spotting," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Calgary, AB, Canada, Apr. 2018, pp. 5474–5478.

[123] R. Menon, H. Kamper, J. Quinn, and T. Niesler, "Fast ASR-free and almost zero-resource keyword spotting using DTW and CNNs for humanitarian monitoring," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Hyderabad, India, Sep. 2018, pp. 2608–2612.

[124] H. Liu, A. Abhyankar, Y. Mishchenko, T. Sénéchal, G. Fu, B. Kulis, N. Stein, A. Shah, and S. N. P. Vitaladevuni, "Metadata-aware end-to-end keyword spotting," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Shanghai, China, Oct. 2020, pp. 2282–2286.

[125] T. Mo, Y. Yu, M. Salameh, D. Niu, and S. Jui, "Neural architecture search for keyword spotting," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Shanghai, China, Oct. 2020, pp. 1982–1986.

[126] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[127] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Represent.*, Toulon, France, Apr. 2017, pp. 1–5.

[128] I. López-Espejo, Z.-H. Tan, and J. Jensen, "Keyword spotting for hearing assistive devices robust to external speakers," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Graz, Austria, Sep. 2019, pp. 3223–3227.

[129] I. López-Espejo, Z.-H. Tan, and J. Jensen, "Improved external speaker-robust keyword spotting for hearing assistive devices," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 28, pp. 1233–1247, 2020.

[130] I. López-Espejo, Z.-H. Tan, and J. Jensen, "A novel loss function and training strategy for noise-robust keyword spotting," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 29, pp. 2254–2266, 2021.

[131] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, Jun. 2017, pp. 4700–4708.

[132] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," in *Proc. ISCA Speech Synthesis Workshop*, Sunnyvale, CA, USA, Sep. 2016, p. 125.

[133] D. C. de Andrade, S. Leo, M. L. Da Silva Viana, and C. Bernkopf, "A neural attention model for speech command recognition," 2018, *arXiv:1808.08929*.

[134] H. Zhou, W. Hu, Y. T. Yeung, and X. Chen, "Energy-friendly keyword spotting system using add-based convolution," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Brno, Czechia, Sep. 2021, pp. 4234–4238.

[135] H. Chen, Y. Wang, C. Xu, B. Shi, C. Xu, Q. Tian, and C. Xu, "AdderNet: Do we really need multiplications in deep learning?" in *Proc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 1468–1477.

[136] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861v1*.

[137] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[138] H. Sundar, J. F. Lehman, and R. Singh, "Keyword spotting in multi-player voice driven games for children," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Dresden, Germany, Sep. 2015, pp. 1660–1664.

[139] Y. Bai, J. Yi, H. Ni, Z. Wen, B. Liu, Y. Li, and J. Tao, "End-to-end keywords spotting based on connectionist temporal classification for Mandarin," in *Proc. Int. Symp. Chin. Spoken Lang. Process.*, Tianjin, China, Oct. 2016, pp. 1–5.

[140] E. Ceolini, J. Anumula, S. Braun, and S.-C. Liu, "Event-driven pipeline for low-latency low-compute keyword spotting and speaker verification system," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Brighton, U.K., May 2019, pp. 7953–7957.

[141] I. Sutskever, O. Vinyals, and Q. Le, "Sequence to sequence learning with neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2014, pp. 3104–3112.

[142] Y.-A. Chung, C.-C. Wu, C.-H. Shen, H.-Y. Lee, and L.-S. Lee, "Audio Word2Vec: Unsupervised learning of audio segment representations using sequence-to-sequence autoencoder," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, San Francisco, CA, USA, Sep. 2016, pp. 765–769.

[143] Z. Liu, T. Li, and P. Zhang, "RNN-T based open-vocabulary keyword spotting in Mandarin with multi-level detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Toronto, ON, Canada, Jun. 2021, pp. 5649–5653.

[144] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, Long Beach, CA, USA, Dec. 2017, pp. 5998–6008.

[145] Z. Zhao and W.-Q. Zhang, "End-to-end keyword search based on attention and energy scorer for low resource languages," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Shanghai, China, Oct. 2020, pp. 2587–2591.

[146] B. Zhang, W. Li, Q. Li, W. Zhuang, X. Chu, and Y. Wang, "AutoKWS: Keyword spotting with differentiable architecture search," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Toronto, ON, Canada, Oct. 2021, pp. 2830–2834.

[147] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, Oct. 1986.

[148] D. Scherer, A. Müller, and S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition," in *Proc. Int. Conf. Artif. Neural Netw.*, Thessaloniki, Greece, Sep. 2010, pp. 92–101.

[149] J. Kiefer and J. Wolfowitz, "Stochastic estimation of the maximum of a regression function," *Ann. Math. Stat.*, vol. 23, no. 3, pp. 462–466, 1952.

[150] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, San Diego, CA, USA, Dec. 2015, pp. 1–5.

[151] B. Wei, M. Yang, T. Zhang, X. Tang, X. Huang, K. Kim, J. Lee, K. Cho, and S.-U. Park, "End-to-end transformer-based open-vocabulary keyword spotting with location-guided local attention," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Brno, Czechia, Aug. 2021, pp. 361–365.

[152] S. An, Y. Kim, H. Xu, J. Lee, M. Lee, and I. Oh, "Robust keyword spotting via recycle-pooling for mobile game," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Graz, Austria, Sep. 2019, pp. 3661–3662.

[153] P. Warden. (2017). *Launching the Speech Commands Dataset*. [Online]. Available: https://ai.googleblog.com/2017/08/launching-speech-commands-dataset.html

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

[155] J. P. A. Pérez, S. Celma, and B. C. López, *Automatic Gain Control: Techniques and Architectures for RF Receivers*. New York, NY, USA: Springer, 2011.

[156] Y. Wang, P. Getreuer, T. Hughes, R. F. Lyon, and R. A. Saurous, "Trainable frontend for robust and far-field keyword spotting," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, New Orleans, LO, USA, May 2017, pp. 5670–5674.

[157] Y. Gu, Z. Du, H. Zhang, and X. Zhang, "A monaural speech enhancement method for robust small-footprint keyword spotting," 2019, *arXiv:1906.08415*.

[158] T. Menne, R. Schlüter, and H. Ney, "Investigation into joint optimization of single channel speech enhancement and acoustic modeling for robust ASR," in *Proc. Int. Conf. Acoust., Speech Signal Process.*, Brighton, U.K., May 2019, pp. 6660–6664.

[159] M. Jung, Y. Jung, J. Goo, and H. Kim, "Multi-task network for noise-robust keyword spotting and speaker verification using CTC-based soft VAD and global query attention," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Shanghai, China, Oct. 2020, pp. 931–935.

[160] Z. Zhang, J. Geiger, J. Pohjalainen, A. E.-D. Mousa, W. Jin, and B. Schuller, "Deep learning for environmentally robust speech recognition: An overview of recent developments," *ACM Trans. Intell. Syst. Technol.*, vol. 9, no. 5, pp. 1–28, Jul. 2018.

[161] K. Wang, B. He, and W.-P. Zhu, "TSTNN: Two-stage transformer based neural network for speech enhancement in the time domain," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Toronto, ON, Canada, Oct. 2021, pp. 7098–7102.

[162] Y. A. Huang, T. Z. Shabestary, A. Gruenstein, and L. Wan, "Multi-microphone adaptive noise cancellation for robust hotword detection," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Graz, Austria, Sep. 2019, pp. 1233–1237.
[163] Google. *Google Nest and Home device specifications*. Accessed: Jan. 7, 2022. [Online]. Available: https://support.google.com/googlenest/answer/7072284?hl=en
[164] N. Ito, N. Ono, E. Vincent, and S. Sagayama, "Designing the Wiener post-filter for diffuse noise suppression using imaginary parts of inter-channel cross-spectra," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Dallas, TX, USA, Oct. 2010, pp. 2818–2821.
[165] S. Lefkimmiatis and P. Maragos, "A generalized estimation approach for linear and nonlinear microphone array post-filters," *Speech Commun.*, vol. 49, pp. 657–666, Oct. 2007.
[166] W. Kellermann, "Beamforming for Speech and Audio Signals," in *Handbook Signal Processing Acoustics*, D. Havelock, S. Kuwano, and M. Vorländer, Eds. New York, NY, USA: Springer, 2008, pp. 691–702.
[167] S.-J. Chen, A. S. Subramanian, H. Xu, and S. Watanabe, "Building state-of-the-art distant speech recognition using the CHiME-4 challenge with a setup of speech enhancement baseline," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Hyderabad, India, Sep. 2018, pp. 1571–1575.
[168] X. Xiao, S. Watanabe, H. Erdogan, L. Lu, J. Hershey, M. L. Seltzer, G. Chen, Y. Zhang, M. Mandel, and D. Yu, "Deep beamforming networks for multi-channel speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Shanghai, China, Mar. 2016, pp. 5745–5749.
[169] T. Bluche and T. Gisselbrecht, "Predicting detection filters for small footprint open-vocabulary keyword spotting," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Shanghai, China, Oct. 2020, pp. 2552–2556.
[170] H.-J. Park, P. Zhu, I. L. Moreno, and N. Subrahmanya, "Noisy student-teacher training for robust keyword spotting," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Brno, Czechia, Aug. 2021, pp. 331–335.
[171] J. Malek and J. Zdansky, "On practical aspects of multi-condition training based on augmentation for reverberation-/noise-robust speech recognition," *Lect. Notes Comput. Sci.*, vol. 11697, pp. 251–263, Oct. 2019.
[172] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," 2019, *arXiv:1904.08779v3*.
[173] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *Proc. Int. Conf. Learn. Represent.*, Banff, CA, Canada, Apr. Apr. 2014, pp. 1–15.
[174] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. Int. Conf. Mach. Learn.*, Lille, France, Jul. 2015, pp. 1–10.
[175] J. Lin, K. Kilgour, D. Roblek, and M. Sharifi, "Training keyword spotters with limited and synthesized speech data," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Barcelona, Spain, May 2020, pp. 7474–7478.
[176] A. Werchniak, R. B. Chicote, Y. Mishchenko, J. Droppo, J. Condal, P. Liu, and A. Shah, "Exploring the application of synthetic audio in training keyword spotters," in *Proc. Int. Conf. Acoust., Speech Signal Process.*, Toronto, ON, Canada, Oct. 2021, pp. 7993–7996.
[177] K. Zhang, Z. Wu, D. Yuan, J. Luan, J. Jia, H. Meng, and B. Song, "Re-weighted interval loss for handling data imbalance problem of end-to-end keyword spotting," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Shanghai, China, Oct. 2020, pp. 2567–2571.
[178] K. Sohn, "Improved deep metric learning with multi-class N-pair loss objective," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, Barcelona, Spain, Dec. 2016, pp. 1857–1865.
[179] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, May 2015, pp. 815–823.
[180] J. Huh, M. Lee, H. Heo, S. Mun, and J. S. Chung, "Metric learning for keyword spotting," in *Proc. Spoken Lang. Technol. Workshop*, Shenzhen, China, Jan. 2021, pp. 133–140.
[181] L. Kaushik, A. Sangwan, and J. H. Hansen, "Automatic audio sentiment extraction using keyword spotting," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Dresden, Germany, Sep. 2015, pp. 2709–2713.
[182] L. Kaushik, A. Sangwan, and J. H. L. Hansen, "Automatic sentiment detection in naturalistic audio," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 25, no. 8, pp. 1668–1679, Aug. 2017.
[183] S. Lee, A. Potamianos, and S. Narayanan, "Acoustics of children's speech: Developmental changes of temporal and spectral parameters," *J. Acoust. Soc. Amer.*, vol. 105, no. 3, pp. 1455–1468, Mar. 1999.
[184] M. Wöllmer, B. W. Schüller, A. Batliner, S. Steidl, and D. Seppi, "Tandem decoding of children's speech for keyword detection in a child-robot interaction scenario," *ACM Trans. Speech Lang. Process.*, vol. 7, pp. 1–22, Oct. 2011.
[185] M. Vacher, B. Lecouteux, and F. Portet, "On distant speech recognition for home automation," *Lect. Notes Comput. Sci.*, vol. 8700, pp. 161–188, Dec. 2015.
[186] M. Rayner, B. A. Hockey, J.-M. Renders, N. Chatzichrisafis, and K. Farrell, "Spoken dialogue application in space: The clarissa procedure browser," in *Speech Technology: Theory Application*, F. Chen and K. Jokinen, Eds. Cham, Switzerland: Springer, 2010, ch. 12, pp. 221–250.
[187] S. Team. (2017). *Hey Siri: An On-device DNN-powered Voice Trigger for Apple's Personal Assistant*. [Online]. Available: https://machinelearning.apple.com/research/hey-siri
[188] Statista. (2020). *Number of Digital Voice Assistants in Use Worldwide From 2019 to 2024*. [Online]. Available: https://www.statista.com/statistics/973815/worldwide-digital-voice-assistant-in-use/
[189] (2021). *Keyword Recognition*. [Online]. Available: https://docs.microsoft.com/en-us/azure/cognitive-services/speech-service/keyword-recognition-overview
[190] V. Garg, W. Chang, S. Sigtia, S. Adya, P. Simha, P. Dighe, and C. Dhir, "Streaming transformer for hardware efficient voice trigger detection and false trigger mitigation," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Brno, Czechia, Aug. 2021, pp. 4209–4213.
[191] C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-24, no. 4, pp. 320–327, Aug. 1976.
[192] J. C. Brown, "Calculation of a constant Q-spectral transform," *J. Acoust. Soc. Amer.*, vol. 89, p. 425, Jan. 1991.
[193] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Brisbane, QC, Australia, Apr. 2015, pp. 5206–5210.
[194] R. Leonard, "A database for speaker-independent digit recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, San Diego, CA, USA, Mar. 1984, pp. 1–5.
[195] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, "DARPA TIMIT acoustic phonetic continuous speech corpus CDROM," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. 4930, 1993.
[196] H. Chen, C.-C. Leung, L. Xie, B. Ma, and H. Li, "Unsupervised bottleneck features for low-resource query-by-example spoken term detection," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, San Francisco, CA, USA, Sep. 2016, pp. 923–927.
[197] D. B. Paul and J. M. Baker, "The design for the wall street journal-based CSR corpus," in *Proc. Speech Natural Lang.*, Harriman, NY, USA, Feb. 1992, pp. 1–5. [Online]. Available: https://www.aclweb.org/anthology/H92-1073
[198] D. Leroy, A. Coucke, T. Lavril, T. Gisselbrecht, and J. Dureau, "Federated learning for keyword spotting," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Brighton, U.K., May 2019, pp. 6341–6345.
[199] J. Du, X. Na, X. Liu, and H. Bu, "AISHELL-2: Transforming Mandarin ASR research into industrial scale," 2018, *arXiv:1808.10583*.
[200] B. Kim, M. Lee, J. Lee, Y. Kim, and K. Hwang, "Query-by-example on-device keyword spotting," in *Proc. IEEE Autom. Speech Recognit. Understand. Workshop*, Singapore, Dec. 2019, pp. 532–538.
[201] Y. Gao, Y. Mishchenko, A. Shah, S. Matsoukas, and S. Vitaladevuni, "Towards data-efficient modeling for wake word spotting," in *Proc. IEEE Int. Conf. Acoust., Speech Singal Process.*, Barcelona, Spain, May 2020, pp. 7479–7483.
[202] M. Chen, S. Zhang, M. Lei, Y. Liu, H. Yao, and J. Gao, "Compact feedforward sequential memory networks for small-footprint keyword spotting," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Hyderabad, India, Sep. 2018, pp. 2663–2667.
[203] H.-G. Hirsch. (2015). *FaNT-Filtering and Noise Adding Tool*. [Online]. Available: https://github.com/i3thuan5/FaNT
[204] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *Proc. Eur. Signal Process. Conf.*, Budapest, Hungary, Aug./Sep. 2016, pp. 1128–1132.
[205] J. Thiemann, N. Ito, and E. Vincent, "The diverse environments multi-channel acoustic noise database (DEMAND): A database of multichannel environmental noise recordings," *Proc. Meetings Acoust.*, vol. 19, no. 1, p. 15, 2013. [Online]. Available: https://asa.scitation.org/doi/abs/10.1121/1.4799597

[206] D. Snyder, G. Chen, and D. Povey, "MUSAN: A music, speech, and noise corpus," 2015, *arXiv:1510.08484*.

[207] A. Varga and H. J. M. Steeneken, "Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech Commun.*, vol. 12, no. 3, pp. 247–251, Jul. 1993.

[208] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third 'CHiME' speech separation and recognition challenge: Dataset, task and baselines," in *Proc. IEEE Autom. Speech Recognit. Understand. Workshop*, Scottsdale, CA, USA, Dec. 2015, pp. 504–511.

[209] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third 'CHiME' speech separation and recognition challenge: Analysis and outcomes," *Comput. Speech Lang.*, vol. 46, pp. 605–626, Nov. 2017.

[210] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, New Orleans, LO, USA, Oct. 2017, pp. 5220–5224.

[211] *Creative Commons Attribution 4.0 International*. Accessed: Jan. 7, 2022. [Online]. Available: https://creativecommons.org/licenses/by/4.0/

[212] *Machine Learning Crash Course—Classification: Accuracy*. Accessed: Jan. 7, 2022. [Online]. Available: https://developers.google.com/machine-learning/crash-course/classification/accuracy

[213] J. P. Mower, "PREP-mt: Predictive RNA editor for plant mitochondrial genes," *BMC Bioinf.*, vol. 6, no. 1, p. 15, Dec. 2005.

[214] N. V. Chawla, *Data Mining for Imbalanced Datasets: An Overview*. Boston, MA, USA: Springer, 2005, pp. 853–867.

[215] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *Int. J. Data Mining Knowl. Manage. Process.*, vol. 5, no. 2, pp. 1–11, 2015.

[216] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann, "The binormal assumption on precision-recall curves," in *Proc. Int. Conf. Pattern Recognit.*, Istanbul, Turkey, Aug. 2010, pp. 4263–4266.

[217] R. Pokrywka, "Reducing false alarm rate in anomaly detection with layered filtering," *Lect. Notes Comput. Sci.*, vol. 5101, pp. 396–404, Oct. 2008.

[218] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006.

[219] H. Benisty, I. Katz, K. Crammer, and D. Malah, "Discriminative keyword spotting for limited-data applications," *Speech Commun.*, vol. 99, pp. 1–11, Oct. 2018.

[220] R. Menon, H. Kamper, E. van der Westhuizen, J. Quinn, and T. Niesler, "Feature exploration for almost zero-resource ASR-free keyword spotting using a multilingual bottleneck extractor and correspondence autoencoders," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Graz, Austria, Sep. 2019, pp. 3475–3479.

[221] J. Wintrode and J. Wilkes, "Fast lattice-free keyword filtering for accelerated spoken term detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Barcelona, Spain, May 2020, pp. 7469–7473.

[222] W. H. Lee, P. D. Gader, and J. N. Wilson, "Optimizing the area under a receiver operating characteristic curve with application to landmine detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 2, pp. 389–397, Feb. 2007.

[223] J. Guo, K. Kumatani, M. Sun, M. Wu, A. Raju, N. Ström, and A. Mandal, "Time-delayed bottleneck highway networks using a DFT feature for keyword spotting," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Calgary, AB, Canada, Apr. 2018, pp. 5489–5493.

[224] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, "The DET curve in assessment of detection task performance," in *Proc. Eur. Conf. Speech Commun. Technol.*, Rhodes, Greece, Sep. 1997, pp. 1895–1898.

[225] R. W. Broadley, J. Klenk, S. B. Thies, L. P. J. Kenney, and M. H. Granat, "Methods for the real-world evaluation of fall detection technology: A scoping review," *MDPI Sens.*, vol. 18, pp. 1–28, Oct. 2018.

[226] D. A. van Leeuwen and N. Brümmer, "An introduction to application-independent evaluation of speaker recognition systems," *Lecture Notes Artif. Intell.*, vol. 4343, pp. 330–353, Dec. 2007.

[227] A. Larcher, K. A. Lee, B. Ma, and H. Li, "Text-dependent speaker verification: Classifiers, databases and RSR2015," *Speech Commun.*, vol. 60, pp. 56–77, Oct. 2014.

[228] Z. Bai and X.-L. Zhang, "Speaker recognition based on deep learning: An overview," 2020, *arXiv:2012.00931*.

[229] A. K. Sarkar, Z.-H. Tan, H. Tang, S. Shon, and J. Glass, "Time-contrastive learning based deep bottleneck features for text-dependent speaker verification," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 27, no. 8, pp. 1267–1279, Aug. 2019.

[230] S.-G. Leem, I.-C. Yoo, and D. Yook, "Multitask learning of deep neural network-based keyword spotting for IoT devices," *IEEE Trans. Consum. Electron.*, vol. 65, no. 2, pp. 188–194, May 2019.

[231] *OpenKWS13 Keyword Search Evaluation Plan*, National Institute of Standards and Technology, Gaithersburg, MD, USA, 2013.

[232] R. Tang, J. Lee, A. Razi, J. Cambre, I. Bicking, J. Kaye, and J. Lin, "Howl: A deployed, open-source wake word detection system," in *Proc. 2nd Workshop NLP Open Source Softw. (NLP-OSS)*, Nov. 2020, pp. 61–65. [Online]. Available: https://www.aclweb.org/anthology/2020.nlposs-1.9

[233] P. Golik, Z. Tüske, R. Schlüter, and H. Ney, "Multilingual features based keyword search for very low-resource languages," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Dresden, Germany, Sep. 2015, pp. 1260–1264.

[234] H. Wang, A. Ragni, M. J. F. Gales, K. M. Knill, P. C. Woodland, and C. Zhang, "Joint decoding of tandem and hybrid systems for improved keyword spotting on low resource languages," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Dresden, Germany, Sep. 2015, pp. 3660–3664.

[235] C.-C. Leung, L. Wang, H. Xu, J. Hou, V. T. Pham, H. Lv, L. Xie, X. Xiao, C. Ni, B. Ma, E. S. Chng, and H. Li, "Toward high-performance language-independent query-by-example spoken term detection for MediaEval 2015: Post-evaluation analysis," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, San Francisco, CA, USA, Sep. 2016, pp. 3703–3707.

[236] G. Huang, A. Gorin, J.-L. Gauvain, and L. Lamel, "Machine translation based data augmentation for Cantonese keyword spotting," in *Proc. Int. Conf. Acoust., Speech Signal Process.*, Shanghai, China, Mar. 2016, pp. 6020–6024.

[237] N. F. Chen, P. V. Tung, H. Xu, X. Xiao, D. V. Hai, C. Ni, I.-F. Chen, S. Sivadas, C.-H. Lee, E. S. Chng, B. Ma, and H. Li, "Exemplar-inspired strategies for low-resource spoken keyword search in Swahili," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Shanghai, China, Mar. 2016, pp. 6040–6044.

[238] J. Trmal, M. Wiesner, V. Peddinti, X. Zhang, P. Ghahremani, Y. Wang, V. Manohar, H. Xu, D. Povey, and S. Khudanpur, "The Kaldi OpenKWS system: Improving low resource keyword search," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Stockholm, Sweden, Aug. 2017, pp. 3597–3601.

[239] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves," in *Proc. Int. Conf. Mach. Learn.*, Pittsburgh, PA, USA, Jun. 2006, pp. 233–240.

[240] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PLoS ONE*, vol. 10, no. 3, pp. 1–21, Mar. 2015.

[241] C. J. van Rijsbergen, *Information Retrieval*. Oxford, U.K.: Butterworth-Heinemann, 1979.

[242] Y. Jianbin, K. Jian, Z. Wei-Qiang, and L. Jia, "Mutitask learning based muti-examples keywords spotting in low resource condition," in *Proc. Int. Conf. Signal Process.*, Beijing, China, Aug. 2018, pp. 581–585.

[243] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," in *Proc. Int. Conf. Learn. Represent.*, Toulon, France, Apr. 2017, pp. 1–5.

[244] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, CA, USA, Jun. 2018, pp. 7132–7141.

[245] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent.*, May 2021, pp. 1–5.

[246] R. Su, X. Liu, L. Wang, and J. Yang, "Cross-domain deep visual feature generation for Mandarin audio–Visual speech recognition," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 28, pp. 185–197, 2020.

[247] T. Makino, H. Liao, Y. Assael, B. Shillingford, B. Garcia, O. Braga, and O. Siohan, "Recurrent neural network transducer for audio-visual speech recognition," in *Proc. Autom. Speech Recognit. Understand. Workshop*, Singapore, Sep. 2019, pp. 905–912.

[248] P. Zhou, W. Yang, W. Chen, Y. Wang, and J. Jia, "Modality attention for end-to-end audio-visual speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Brighton, U.K., May 2019, pp. 6565–6569.

[249] J. Yu, S.-X. Zhang, J. Wu, S. Ghorbani, B. Wu, S. Kang, S. Liu, X. Liu, H. Meng, and D. Yu, "Audio-visual recognition of overlapped speech for the LRS2 dataset," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Barcelona, Spain, May 2020, pp. 6984–6988.
[250] P. Wu, H. Liu, X. Li, T. Fan, and X. Zhang, "A novel lip descriptor for audio-visual keyword spotting based on adaptive decision fusion," *IEEE Trans. Multimedia*, vol. 18, no. 3, pp. 326–338, Mar. 2016.
[251] R. Ding, C. Pang, and H. Liu, "Audio-visual keyword spotting based on multidimensional convolutional neural network," in *Proc. IEEE Int. Conf. Image Process.*, Athens, Greece, Aug. 2018, pp. 4138–4142.
[252] L. Momeni, T. Afouras, T. Stafylakis, S. Albanie, and A. Zisserman, "Seeing wake words: Audio-visual keyword spotting," in *Proc. Brit. Mach. Vis. Virtual Conf.*, Sep. 2020, pp. 1–13.
[253] J.-S. Lee and C. H. Park, *Adaptive Decision Fusion for Audio-Visual Speech Recognition*. London, U.K.: IntechOpen, 2008, pp. 275–296.
[254] J. S. Chung and A. Zisserman, "Lip reading in the wild," in *Proc. Asian Conf. Comput. Vis.*, Taipei, Taiwan, Nov. 2016, pp. 1–8.
[255] J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman, "Lip reading sentences in the wild," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, Jul. 2017, pp. 3444–3453.
[256] T. Afouras, J. Son Chung, and A. Zisserman, "LRS3-TED: A large-scale dataset for visual speech recognition," 2018, *arXiv:1809.00496*.
[257] T. Parcollet, M. Morchid, and G. Linarès, "E2E-SINCNET: Toward fully end-to-end speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Barcelona, Spain, May 2020, pp. 7714–7718.
[258] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2014, pp. 1–9.
[259] Y. Ouali, C. Hudelot, and M. Tami, "An overview of deep semi-supervised learning," 2020, *arXiv:2006.05278*.
[260] S. Sigtia, E. Marchi, S. Kajarekar, D. Naik, and J. Bridle, "Multi-task learning for speaker verification and voice trigger detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Barcelona, Spain, May 2020, pp. 6844–6848.
[261] Y. Jia, X. Wang, X. Qin, Y. Zhang, X. Wang, J. Wang, D. Zhang, and M. Li, "The 2020 personalized voice trigger challenge: Open datasets, evaluation metrics, baseline system and results," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Brno, Czechia, Aug. 2021, pp. 4239–4243.
[262] I. López-Espejo, A. M. Peinado, A. M. Gomez, and J. A. Gonzalez, "Dual-channel spectral weighting for robust speech recognition in mobile devices," *Digit. Signal Process.*, vol. 75, pp. 13–24, Apr. 2018.

IVÁN LÓPEZ-ESPEJO received the M.Sc. degree in telecommunications engineering, the M.Sc. degree in electronics engineering, and the Ph.D. degree in information and communications technology from the University of Granada, Granada, Spain, in 2011, 2013, and 2017, respectively. In 2018, he was the Leader of the Speech Technology Team, Veridas, Pamplona, Spain. Since 2019, he has been a Postdoctoral Researcher with the Section for Artificial Intelligence and Sound, Department of Electronic Systems, Aalborg University, Aalborg, Denmark. His research interests include speech enhancement and robust speech recognition, multi-channel speech processing, and speaker verification.

ZHENG-HUA TAN (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering from Hunan University, Changsha, China, in 1990 and 1996, respectively, and the Ph.D. degree in electronic engineering from Shanghai Jiao Tong University (SJTU), Shanghai, China, in 1999.

He was a Visiting Scientist at the Computer Science and Artificial Intelligence Laboratory, MIT, Cambridge, MA, USA, an Associate Professor at SJTU, and a Postdoctoral Fellow at the KAIST, Daejeon, South Korea. He is currently a Professor with the Department of Electronic Systems and the Co-Head of the Centre for Acoustic Signal Processing Research, Aalborg University, Aalborg, Denmark. He has (co)authored over 200 refereed publications. His research interests include machine learning, deep learning, pattern recognition, speech and speaker recognition, noise-robust speech processing, multimodal signal processing, and social robotics. He has served as an Editorial Board Member for Computer Speech and Language and was a Guest Editor for the IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING and *Neurocomputing*. He was the General Chair of IEEE MLSP 2018 and a TPC Co-Chair of IEEE SLT 2016. He is the Chair of the IEEE Signal Processing Society Machine Learning for Signal Processing Technical Committee (MLSP TC). He is an Associate Editor of the IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH AND LANGUAGE PROCESSING.

JOHN H. L. HANSEN (Fellow, IEEE) received the B.S.E.E. degree from the College of Engineering, Rutgers University, New Brunswick, NJ, USA, the M.S. and Ph.D. degrees in electrical engineering from the Georgia Institute of Technology, Atlanta, GA, USA, and the Doctor Technices Honoris Causa degree (Hons.) from Aalborg University, Aalborg, Denmark, in 2016, in recognition of his contributions to the field of speech signal processing and speech or language or hearing sciences. From 1998 to 2005, he was the Department Chair and a Professor of speech, language, and hearing sciences, and a Professor of electrical and computer engineering with the University of Colorado Boulder, Boulder, CO, USA, where he co-founded and was an Associate Director of the Center for Spoken Language Research. In 1988, he established the Robust Speech Processing Laboratory. In 2005, he joined the Erik Jonsson School of Engineering and Computer Science, The University of Texas at Dallas, Richardson, TX, USA, where he is currently an Associate Dean for research and a Professor of electrical and computer engineering. He also holds the Distinguished University Chair in telecommunications engineering and a joint appointment as a Professor of speech and hearing with the School of Behavioral and Brain Sciences. From 2005 to 2012, he was the Head of the Department of Electrical Engineering, The University of Texas at Dallas. At UT Dallas, he established the Center for Robust Speech Systems. He has supervised 92 Ph.D. or M.S. thesis students, which include 51 Ph.D. and 41 M.S. or M.A. He has authored or coauthored 765 journals and conference papers, including 13 textbooks in the field of speech processing and language technology, signal processing for vehicle systems. He is the coauthor of the textbook *Discrete-Time Processing of Speech Signals* (IEEE Press, 2000), *Vehicles, Drivers and Safety: Intelligent Vehicles and Transportation* (vol. 2 DeGruyter, 2020), and *Digital Signal Processing for In-Vehicle Systems and Safety* (Springer, 2012), and the Lead Author of *The Impact of Speech Under 'Stress' on Military Speech Technology* (NATO RTO-TR-10, 2000). His research interests include machine learning for speech and language processing, speech processing, analysis, and modeling of speech and speaker traits, speech enhancement, signal processing for hearing impaired or cochlear implants, machine learning-based knowledge estimation and extraction of naturalistic audio, and in-vehicle driver modeling and distraction assessment for human–machine interaction. He is an IEEE Fellow for contributions to robust speech recognition in stress and noise, and ISCA Fellow for contributions to research for speech processing of signals under adverse conditions. He was a recipient of the 2020 Provost's Award for Excellence in Graduate Student Supervision from The University of Texas at Dallas and the 2005 University of Colorado Teacher Recognition Award. He was a recipient of Acoustical Society of America's 25 Year Award in 2010, and is also serving as the ISCA President (2017–2022). He is also

a member and the Past Vice-Chair on U.S. Office of Scientific Advisory Committees (OSAC) for OSAC-Speaker in the voice forensics domain, from 2015 to 2021. He organized and was the General Chair of ISCA Interspeech-2002, a Co-Organizer and a Technical Program Chair of the IEEE ICASSP-2010, Dallas, TX, USA, and a Co-Chair and an Organizer of IEEE SLT-2014, Lake Tahoe, NV, USA. He will be the Technical Program Chair of the IEEE ICASSP-2024, and a Co-Chair and an Organizer of ISCA INTERSPEECH-2022. He was the IEEE Technical Committee (TC) Chair and a member of the IEEE Signal Processing Society: Speech-Language Processing Technical Committee (SLTC), from 2005 to 2008, and from 2010 to 2014, elected the IEEE SLTC Chairperson, from 2011 to 2013, and elected an ISCA Distinguished Lecturer, from 2011 to 2012. He was a member of the IEEE Signal Processing Society Educational Technical Committee, from 2005 to 2010, a Technical Advisor to the U.S. Delegate for NATO (IST/TG-01), an IEEE Signal Processing Society Distinguished Lecturer, from 2005 to 2006, an Associate Editor of the IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, from 1992 to 1999, and the IEEE SIGNAL PROCESSING LETTERS, from 1998 to 2000, an Editorial Board Member of the *IEEE Signal Processing Magazine*, from 2001 to 2003, and the Guest Editor in October 1994 for Special Issue on Robust Speech Recognition for the IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING. He is also an Associate Editor for the *JASA*, and was on the Speech Communications Technical Committee for Acoustical Society of America, from 2000 to 2003.

JESPER JENSEN received the M.Sc. degree in electrical engineering and the Ph.D. degree in signal processing from Aalborg University, Aalborg, Denmark, in 1996 and 2000, respectively. From 1996 to 2000, he was with the Center for Person Kommunikation (CPK), Aalborg University, as a Ph.D. Student and an Assistant Research Professor. From 2000 to 2007, he was a Postdoctoral Researcher and an Assistant Professor with the Delft University of Technology, Delft, The Netherlands, and an External Associate Professor with Aalborg University. He is currently a Senior Principal Scientist with Oticon A/S, Copenhagen, Denmark, where his main responsibility is scouting and development of new signal processing concepts for hearing aid applications. He is also a Professor with the Section for Artificial Intelligence and Sound (AIS), Department of Electronic Systems, Aalborg University. He is also a Co-Founder of the Centre for Acoustic Signal Processing Research (CASPR), Aalborg University. His main research interests include acoustic signal processing, including signal retrieval from noisy observations, coding, speech and audio modification and synthesis, intelligibility enhancement of speech signals, signal processing for hearing aid applications, and perceptual aspects of signal processing.