Error-Diffusion Based Speech Feature Quantization for Small-Footprint Keyword Spotting

Mengjie Luo, Dingyi Wang, Xiaoqin Wang, Shushan Qiao, and Yumei Zhou

***Abstract*—Neural network based keyword spotting (KWS) system is a critical component for user interaction in current smart devices. Although small-footprint networks have been widely explored to reduce deployment overhead, low-precision input feature representation still lacks in-depth research. In this letter, an error-diffusion based speech feature quantization method is proposed. Specifically, our algorithm adapts image processing to quantize the input speech feature maps in arbitrary bits. Experiments show that in the 10-keyword KWS task, our 3-bit representation only brings a 0.45% average accuracy drop compared to the full-precision log-Mel spectrograms while others drop over 3%. In the 2 keywords task, our 3-bit representation produces no significant differences, while 1-bit quantization only leads to an average of 1.7% accuracy drop and is even capable of handling similar keywords and imbalanced data distribution. The result proves our method, to the best of our knowledge, is the first practical method that supports as low as 1-bit quantization for single-channel speech features in small-footprint KWS. In addition, we analyze the impact of error-diffusion directions and conclude that time-direction diffusion is more suitable for temporal convolutional networks.**

***Index Terms*—Keyword spotting, speech feature quantization, error diffusion, image processing, convolutional neural networks.**

I. Introduction

KEYWORD spotting (KWS), a classification task aiming to detect single-word or phrase commands from an audio stream, has gained much attention in recent years. With the success of deep learning in a variety of cognitive task, neural network based approaches [1]–[6] have become popular for KWS and presented convincing performance. Specifically, convolutional neural networks (CNNs) based KWS [6]–[11] show remarkable accuracy [12]. However, such networks require considerable computations, making it difficult for resource-limited device deployment.

Recently, several small-footprint network architectures have been introduced in KWS to resolve the computational drawbacks of conventional CNN, such as DSCNN [13], TENet [14], TC-RESNet [15], MatchboxNet [16]. Moreover, low-precision network quantization techniques [17] have been widely explored for reduced memory overhead and efficient inference. In particular, extreme compact KWS have been implemented [18], [19] using binary weight neural network (BWN) [20], [21] architectures. However, these studies focus only on the optimization of neural network itself while still using high precision speech features which lead to extra computational cost.

Therefore, researchers start to look for low-precision speech representation approaches to further diminish the energy consumption and memory footprint of KWS. [12] first indicates that the design of new extremely-light and compact features in small-footprint KWS systems can be promising for further study. For example, [22] proposes a log-Mel based linear quantizer that supports as low as 2 bits with a cost of insignificant accuracy loss on a relatively complex network EdgeSpeechNet-A[23]. However, this work hasn't demonstrated effectiveness on small-footprint networks. Besides, its use of 40 Mel filters may help compensate for the quantization loss, as [24] indicates that much of the spectral information is redundant in the field of KWS.

In this letter, we aim to find a method that supports as low as 1-bit quantization while maintaining high accuracy in small-footprint KWS applications. Different from the prior works which use the power variation of speech for extreme low precision quantization [22], [25], we solve the problem from the perspective of preserving the image quality of speech feature maps. The proposed method adapts error diffusion algorithm to quantize speech feature maps or even binarize them directly, which makes the deployment of a fully binary neural network (BNN) possible. To the best of our knowledge, it is the first attempt to apply an image processing method for speech feature quantization. Experiments show that the proposed low precision features are effective in a variety of KWS tasks, yielding higher recognition accuracy than linearly-quantized [22] and standard max-min quantization for different CNN architectures. Furthermore, we discuss the impact of error diffusion directions on recognition accuracy.

II. Proposed Method

A. Error-Diffusion Based Speech Feature Quantization

Error-diffusion is a well-known image processing algorithm to quantize a continuous tone image (e.g., grayscale and color image) into a halftone one and keep as much of the perceived detail as possible. Considering that the feature maps in CNN based KWS systems can be treated as a set of images, the quantization approaches of the image could be used here to preserve feature

Manuscript received April 6, 2022; revised May 15, 2022; accepted May 23, 2022. Date of publication May 30, 2022; date of current version June 22, 2022. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Xiaodong Cui. *(Corresponding author: Xiaoqin Wang.)*

Mengjie Luo is with the Institute of Microelectronics of the Chinese Academy of Sciences, Beijing 100029, China, and also with the University of Chinese Academy of Sciences, Beijing 100029, China (e-mail: luomengjie@ime.ac.cn).

Dingyi Wang is with the Institute of Microelectronics of the Chinese Academy of Sciences, Beijing 100029, China (e-mail: wangdingyi@ime.ac.cn).

Xiaoqin Wang, Shushan Qiao, and Yumei Zhou are with the Institute of Microelectronics of the Chinese Academy of Sciences, Beijing 100029, China, and also with the University of Chinese Academy of Sciences, Beijing 100029, China (e-mail: wangxiaoqin@ime.ac.cn; qiaoshushan@ime.ac.cn; ymzhou@ime.ac.cn).

Digital Object Identifier 10.1109/LSP.2022.3179208

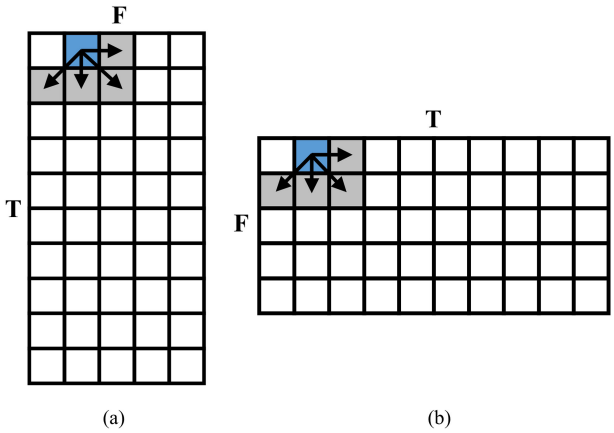

Fig. 1. Direction of error diffusion. (a) Time-Direction. (b) Filter-Direction.

maps' information. Therefore, our algorithm adapts a famous error-diffusion method called Floyd-Steinberg dithering [26] and applies it into the quantization of speech feature in KWS, it takes advantage of the error-diffusion algorithm to maintain the quality of the quantized feature maps and extend the precision representation to arbitrary bits (including binary).

The quantization basis we select here is the log-Mel (filterbanks) spectrogram, as [12] proved Mel-scale-related features are still among the best choices for KWS tasks. Its conclusion is derived from the fact that multiple attempts [27]–[29] for a learnable speech feature turned out to have similar results as log-Mel spectrograms.

The essence of the algorithm is to take the quantization error as a result of thresholding one pixel and diffuse to its neighbors where it influences their threshold operation. Starting from the top-left pixel, the algorithm runs firstly along the width of feature maps and then along the height direction. Therefore, the height and width setting corresponding to different dimensions of the Mel-log spectrogram results in two approaches to diffuse quantization errors in practice, we call them Time-Direction (TD) and Filter-Direction (FD) diffusion. Fig. 1 shows the difference of width-height definition between TD and FD diffusion. In this particular example, the time bin of the feature map(T) is 10 and the filter banks(F) is 5. It can be observed in Fig. 1 that the quantization error will be mainly diffused to adjacent filter banks of following time bin in TD while same filter bank of adjacent time bins in FD.

In detail, the algorithm of error-diffusion based speech feature quantization is given in Algorithm 1. For TD diffusion, parameter *H* is the number of time bins and *W* is the number of filter banks, while vice versa for FD diffusion.

Algorithm 1: Error-Diffusion Based Feature Quantization.

Input: $P_{acc}(i,j) \in [0,1]$ is the accumulated value which starts off as the input feature maps. n is the target precision bit width. *W* is the width of feature maps while *H* is heights.

Output: $P_{out}(i,j)$ is the target-precision value of feature maps.

$P_{acc}(i,j) \leftarrow P_{acc}(i,j) \times (2^n - 1)$
for $i \leftarrow 1$ to H **do**
  for $j \leftarrow 1$ to W **do**
    if $(binary\_mode)$ **then**
      if $(P_{acc}(i,j) > 0.5)$ **then**
        $P_{out}(i,j) \leftarrow 1$
      else
        $P_{out}(i,j) \leftarrow 0$
      end if
    else
      $P_{out}(i,j) \leftarrow floor(P_{acc}(i,j))$
    end if
    $e \leftarrow P_{acc}(i,j) - P_{out}(i,j)$
    if $(j < W)$ **then**
      $P_{acc}(i,j+1) \leftarrow P_{acc}(i,j+1) + \frac{7}{16}e$
      end if
    if $(i < H)$ *and* $(j > 1)$ **then**
      $P_{acc}(i+1,j-1) \leftarrow P_{acc}(i+1,j-1) + \frac{3}{16}e$
    end if
    if $(i < H)$ *and* $(j < W)$ **then**
      $P_{acc}(i+1,j) \leftarrow P_{acc}(i+1,j) + \frac{5}{16}e$
      $P_{acc}(i+1,j+1) \leftarrow P_{acc}(i+1,j+1) + \frac{1}{16}e$
    end if
  end for
end for

TABLE I
THE PSNR FOR DIFFERENT METHOD IN 2, 3, 48-BIT REPRESENTATION

Precision \ Method	Error-diffusion	Linearly-quantized	Standard max-min
2	**17.65**	15.06	14.69
3	**26.03**	22.71	21.62
4	**32.90**	30.89	28.33
8	**57.39**	55.36	52.75

B. Speech Feature Maps Evaluation

An example of float log-Mel spectrogram and different 3-bit quantization representations of word "happy" is illustrated in Fig. 2. The loss of image characteristics is obvious in red blocks.

As the speech feature maps are considered as images here, the Peak Signal-to-noise Ratio (PSNR) is used to evaluate the quality of low-precision representations after different quantization methods. The input speech signal is selected from Google Speech Commands Dataset [30] (GSCD) in which each utterance is one-second long.

In PSNR evaluation, the full-precision log-Mel spectrogram is treated as the original image, and the examined feature maps are representations quantized to 8, 4, 3 and 2 bits by different methods. The results in Table I demonstrate that the error-diffusion method produces the highest quality of image at low quantization precision compared to linearly-quantitated representation [22] and the standard max-min quantization.

III. EXPERIMENTAL RESULTS

A. Implementation Details

Dataset: We use recognition accuracy of the KWS task as the main metric to evaluate how well the quantization method performs. Google Speech Commands Dataset is used in the following experiments, it contains about 65k one-second-long

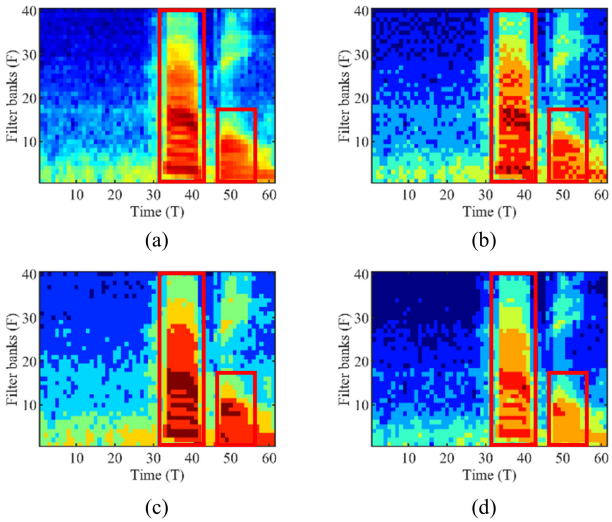

Fig. 2. log-Mel spectrogram of word "happy" in different quantization method: (a) 64-bit full precision. (b) 3-bit error-diffusion. (c) 3-bit linearly-quantized. (d) 3-bit standard max-min.

utterance files of 30 target categories. Besides, we split the experiment into two KWS tasks. In the first task we follow Google's implementation to distinguish 10 keywords with 2 filler classes, including "yes", "no", "up", "down", "left", "right", "on", "off", "stop", "go", silence, and unknown using 3-bit representation. Meanwhile, a 2-keyword task ("happy" and "stop") is used to simulate a small-footprint KWS application where 3-bit and binary representation are examined. Moreover, experiments with similar keywords ("no" and "go") and imbalanced data distribution are conducted to assess the robustness of the proposed method under more challenging scenario. In all tasks, the dataset is split into training, validation, and test sets, with 80% training, 10% validation, and 10% test, respectively.

Data augmentation and preprocessing: Following Google's preprocessing procedures, we apply data augmentations including random shift and noise addition in training phase. Background noises from the dataset are sampled and added with a random proportion following a uniform distribution U(0, 0.1). After that, the signal is time shifted by t seconds and padded with zeros to 1 second, t is sampled from U(-0.1, 0.1).

Experimental Setting: To simulate an edge-device-friendly KWS system, a small-footprint log-Mel feature map is used here, the size of which is 61x10. We apply 10 Mel filters with a 32ms window size and 16ms frame shift, such setting can be seen in several small-footprint KWS deployments [13], [19].

Furthermore, to verify the generality of the proposed method under small-footprint KWS, a set of compact CNNs including CNN-M [13], DS-CNN-S [13], TENet6 [14] and TC-RESNet8 [15] are selected as backbone networks. All the models are trained for 50 epochs with a batch size of 100, and the hyperparameters of each model are set to the same as the corresponding papers.

B. Evaluation Results

1) Low-Precision Log-Mel Spectrogram: Experimental results in 10-keyword KWS task of the 64-bit full precision and 3-bit quantized are summarized in Table II. The accuracy of full-precision log-Mel is treated as the baseline for the quantization methods. As observed, the proposed method can achieve the best accuracy on all models. The average loss of our method on four models is 0.45%, while linearly-quantitated representation and the standard max-min quantization dropped by 3.82% and 4.19%, respectively.

TABLE II
THE ACCURACY OF 10 KEYWORDS KWS IN 3-BIT REPRESENTATION OF DIFFERENT METHODS

Model \ Method	64-bit float log-Mel	3-bit TD diffusion quantized	3-bit linearly quantized	3-bit max-min quantized
CNN-M	92.84%	**92.50%**	89.50%	89.45%
DS-CNN-S	94.29%	**93.58%**	89.23%	89.52%
TENET6	95.25%	**94.96%**	91.82%	90.86%
TC-RESNET8	94.22%	**93.76%**	90.78%	90.01%
Average loss	-	**0.45%**	3.82%	4.19%

TABLE III
THE ACCURACY OF 2 KEYWORDS KWS IN 3-BIT REPRESENTATION OF DIFFERENT METHODS

Model \ Method	64-bit float log-Mel	3-bit TD diffusion quantized	3-bit linearly quantized	3-bit max-min quantized
CNN-M	98.30%	**98.16%**	97.71%	96.32%
DS-CNN-S	98.51%	**98.50%**	97.42%	97.44%
TENET6	98.93%	**98.71%**	98.32%	98.08%
TC-RESNET8	98.59%	**98.38%**	98.23%	97.87%
Average loss	-	**0.15%**	0.66%	1.16%

TABLE IV
COMPARISON OF FLOAT LOG-MEL AND BINARY REPRESENTATION IN 2 KEYWORDS ("HAPPY" AND "STOP") KWS

Model	Acc (%)			Conv1 Ops[ab]		
	Float Rep.	Binary Rep.	Acc loss	Float Rep.	Binary Rep.	Ops Saving
CNN-M	98.30	96.41	1.89	1.8M	0.9M	~2.0x
DS-CNN-S	98.51	96.75	1.76	0.8M	0.4M	~2.0x
TENET6	98.93	97.37	1.56	117.1K	63.1K	~1.9x
TC-RESNET8	98.59	97.02	1.57	58.6K	33.8K	~1.7x
Average	-	-	1.70	-	-	~1.9x

[a]Conv1 Ops: Operations in first convolutional layer
[b]Error diffusion requires 4.46 K extra operations

For 2-keyword KWS task, the proposed method still achieves the highest accuracy as illustrated in Table III. With an average accuracy loss of 0.15%, we can conclude that for 2-keyword small-footprint KWS task, our 3-bit representation has almost no affection on the performance of CNNs.

2) Binary Log-Mel Spectrogram: Firstly, a 2-keyword experiment is conducted using binary spectrogram to evaluate the tradeoff between computational operations (Ops) and accuracy.

A direct advantage of using binary input features is that the multiply-accumulate operations (MACs) could be directly simplified as accumulate operations. As shown in Table IV, the binary representation only leads to an average of 1.7% accuracy drop compared to float spectrogram while reduces about half of computational cost in the first convolution layer.

Besides, to assess the robustness of the proposed method, similar keywords "no" and "go" are used and further examined with extremely imbalanced distribution (no:500, go:3000). As illustrated in Table V, under such difficult condition, both float

TABLE V
THE ACCURACY (%) OF 2 KEYWORDS FOR BINARY REPRESENTATION UNDER CHALLENGING SCENARIO

Model	Float Rep.			Binary Rep.		
	happy stop Bal.	no go Bal.	no go Imb.	happy stop Bal.	no go Bal.	no go Imb.
CNN-M	98.30	96.53	96.34	96.41	94.13	93.32
DS-CNN-S	98.51	97.31	96.46	96.75	94.51	92.82
TENET6	98.93	97.54	97.20	97.37	94.80	94.06
TC-RESNET8	98.59	97.43	97.03	97.02	94.45	93.80
Average loss	-	1.38	1.83	-	2.43	3.39

TABLE VI
THE ACCURACY OF 2 KEYWORDS IN BINARY REPRESENTATION

Model \ Method	64-bit float log-Mel	TD-diffusion quantized	FD-diffusion quantized
CNN-M	98.30%	96.41%	96.34%
DS-CNN-S	98.51%	96.75%	96.74%
TENET6	98.93%	97.37%	95.78%
TC-RESNET8	98.59%	97.02%	95.80%
2D CNN [a] loss	-	**1.83%**	**1.87%**
1D TCNN [b] loss	-	**1.57%**	**2.97%**

[a] 2D CNN: CNN-M and DS-CNN-S.
[b] 1D TCNN: TENET6 and TC-RESNET8.

and binary representation face accuracy drop. Although the 1-bit representation has a bit higher average loss than float log-Mel, it still gets a relatively practical performance (>92%) and proves that the neural networks can learn and classify well using our binary representation.

3) Error-Diffusion Direction: To gain further insight into the impact of the different diffusion direction on KWS accuracy, we use 1-bit representation with TD and FD diffusion on the 2-keyword task to compare two approaches. The experiment results in Table VI show that TD-diffusion brings 1.4% higher accuracy than FD-diffusion in 1D temporal convolutional networks, on the other hand, the difference brought by diffusion directions is insignificant in traditional 2D convolutional networks.

Here we provide a rough explanation for this phenomenon by counting the out-of-kernel pixels that will diffuse error to the data involved in one time of convolution. Fig. 3 gives an example of the effect of two diffusion directions on different convolution kernels. As shown in Fig. 3, the in-kernel data within a 2D traditional CNN typically receives error diffused by pixels around 3 directions in both diffusion approaches. For 1D temporal convolution, on the other hand, in-kernel data only gets affected by pixels from one side in TD diffusion, which is doubled in FD diffusion.

As shown in Fig. 4, assuming the data used for one convolution in a 1D temporal convolution requires $t \times F$ pixels where F is the number of filter banks, while that of a 2D CNN includes $t' \times f$ pixels. The number of error-diffusing pixels N can be formulated as

$$N_{T-1D} = F \tag{1}$$

$$N_{F-1D} = 2F - 1 \tag{2}$$

$$N_{T-2D} = 2t' + f + 1 \tag{3}$$

$$N_{F-2D} = 2f + t' + 1 \tag{4}$$

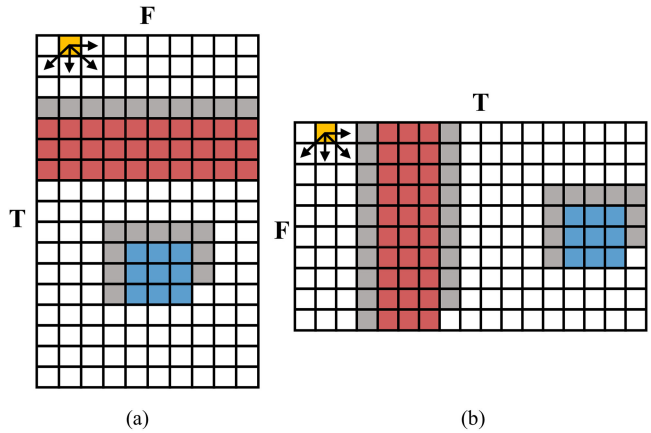

Fig. 3 Impact of error-diffusion direction on two convolution processes. Yellow blocks demonstrate the diffusion direction. Red blocks represent data involved in a 1D temporal convolution. Blue blocks represent data involved in a 2D traditional convolution. Gray blocks represent error-diffusing pixels outside the convolution kernels. (a) TD-diffusion. (b) FD-diffusion.

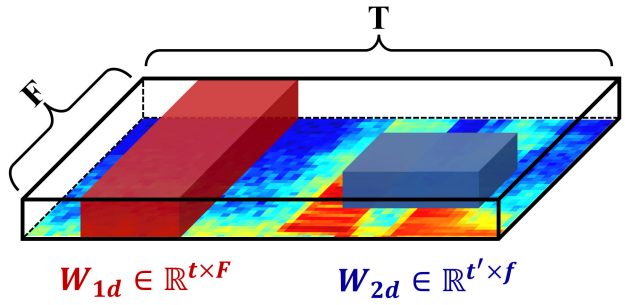

Fig. 4. Example of 1D and 2D kernels.

The subscript T and F denote TD and FD diffusion separately. We can easily derive that for 1D temporal CNNs, the difference in error-diffusing pixels between two directions N_{T-1D} and N_{F-1D} is always $F-1$, while for 2D traditional CNNs is $f - t'$, which is typically smaller than $F-1$ as 2D CNNs tend to use more square kernels. Therefore, we conclude that the make TD-diffusion a better choice for 1D temporal convolutional networks.

IV. CONCLUSION

In this letter, we introduce a novel speech feature quantization approach for KWS based on error-diffusion. The strength of the proposed method lies in two aspects. Firstly, it supports quantization with arbitrary precision as low as 1-bit which is directly interfaceable with binary or ternary neural networks. Secondly, the proposed method reduces the information loss during quantization by preserving the image quality of feature maps. Experiments show that our method produces features with the best PSNR in a variety of precisions. The recognition accuracies of different networks demonstrate that the proposed feature not only has the best performance in small-footprint KWS tasks compared to other low-precision features, but is also robust enough for challenging scenarios like similar-keywords classification and imbalanced training datasets. Moreover, we find that TD diffusion could be more suitable for 1D temporal convolutional networks and discuss about the phenomenon.

REFERENCES

[1] G. Chen, C. Parada, and G. Heigold, "Small-footprint keyword spotting using deep neural networks," in *Proc. ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process.*, 2014, pp. 4087–4091, doi: 10.1109/ICASSP.2014.6854370.

[2] M. Sun *et al.*, "Max-pooling loss training of long short-term memory networks for small-footprint keyword spotting," in *Proc. IEEE Workshop Spoken Lang. Technol.*, 2016, pp. 474–480, doi: 10.1109/SLT.2016.7846306.

[3] S. O. Arik *et al.*, "Convolutional recurrent neural networks for small-footprint keyword spotting," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2017, pp. 1606–1610, doi: 10.21437/Interspeech.2017-1737.

[4] T. N. Sainath and C. Parada, "Convolutional neural networks for small-footprint keyword spotting," in *Proc. Interspeech*, 2015, pp. 1478–1482, doi: 10.21437/Interspeech.2015-352.

[5] R. Kumar, V. Yeruva, and S. Ganapathy, "On convolutional LSTM modeling for joint wake-word detection and text dependent speaker verification," in *Proc. Interspeech*, 2018, pp. 1121–1125, doi: 10.21437/Interspeech.2018-1759.

[6] 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.*, 2018, pp. 5474–5478, doi: 10.1109/ICASSP.2018.8462346.

[7] M. Yu, X. Ji, B. Wu, D. Su, and D. Yu, "End-to-end multi-look keyword spotting," in *Proc. Interspeech*, 2020, pp. 66–70, doi: 10.21437/Interspeech.2020-1521.

[8] 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. INTERSPEECH*, 2020, pp. 2277–2281, doi: 10.21437/Interspeech.2020-1003.

[9] 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.*, 2018, pp. 5479–5483, doi: 10.1109/ICASSP.2018.8461624.

[10] G. Chen, O. Yilmaz, J. Trmal, D. Povey, and S. Khudanpur, "Using proxies for OOV keywords in the keyword search task," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, 2013, pp. 416–421, doi: 10.1109/ASRU.2013.6707766.

[11] A. Riviello, "Binary neural networks for keyword spotting tasks," M.S. thesis, Département de génie électrique, Polytechnique Montréal, Québec, Canada, 2020. [Online]. Available: https://publications.polymtl.ca/5449/

[12] I. López-Espejo, Z.-H. Tan, J. Hansen, and J. Jensen, "Deep spoken keyword spotting: An overview," *IEEE Access*, vol. 10, pp. 4169–4199, 2022, doi: 10.1109/access.2021.3139508.

[13] Y. Zhang, N. Suda, L. Lai, and V. Chandra, "Hello edge: Keyword spotting on microcontrollers," *IEEE J. Solid-State Circuits*, vol. 56, no. 1, pp. 151–164, Nov. 2017. [Online]. Available: http://arxiv.org/abs/1711.07128

[14] X. Li, X. Wei, and X. Qin, "Small-Footprint keyword spotting with multi-scale temporal convolution," in *Proc. Interspeech*, 2020, pp. 1987–1991, doi: 10.21437/Interspeech.2020-3177.

[15] S. Choi *et al.*, "Temporal convolution for real-time keyword spotting on mobile devices," *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, 2019, pp. 3372–3376, doi: 10.21437/Interspeech.2019-1363.

[16] S. Majumdar and B. Ginsburg, "MatchboxNet: 1D time-channel separable convolutional neural network architecture for speech commands recognition," in *Proc. Interspeech*, 2020, pp. 3356–3360, doi: 10.21437/Interspeech.2020-1058.

[17] R. Alvarez, R. Prabhavalkar, and A. Bakhtin, "On the efficient representation and execution of deep acoustic models," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, 2016, pp. 2746–2750, doi: 10.21437/Interspeech.2016-128.

[18] B. Liu *et al.*, "A 22nm, 10.8 μ W/15.1 μ W dual computing modes high power-performance-area efficiency domained background noise aware keyword- spotting processor," *IEEE Trans. Circuits Syst. I: Reg. Papers*, vol. 67, no. 12, pp. 4733–4746, Dec. 2020, doi: 10.1109/TCSI.2020.2997913.

[19] W. Shan *et al.*, "A 510-nW wake-up keyword-spotting chip using serial-FFT-based MFCC and binarized depthwise separable CNN in 28-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 56, no. 1, pp. 151–164, Jan. 2021, doi: 10.1109/JSSC.2020.3029097.

[20] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or -1," Mar. 2016, *arXiv:1602.02830*.

[21] B. Leibe, J. Matas, N. Sebe, and M. Welling, "XNOR-Net: Imagenet classification using binary convolutional neural networks," in *Lecture Notes in Computer Science (Including Subseries Lecture Notes Artificial Intelligence Lecture Notes Bioinformatics)*, vol. 9906, 2016, pp. 7–9, doi: 10.1007/978-3-319-46493-0.

[22] A. Riviello and J. P. David, "Binary speech features for keyword spotting tasks," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, 2019, pp. 3460–3464, doi: 10.21437/Interspeech.2019-1877.

[23] Z. Q. Lin, A. G. Chung, and A. Wong, "EdgeSpeechNets: Highly efficient deep neural networks for speech recognition on the edge," Nov. 2018, *arXiv:1810.08559*.

[24] I. López-Espejo, Z. H. Tan, and J. Jensen, "Exploring filterbank learning for keyword spotting," in *Proc. Eur. Signal Process. Conf.*, 2021, pp. 331–335, doi: 10.23919/Eusipco47968.2020.9287772.

[25] K. Kumar, C. Kim, and R. M. Stern, "Delta-spectral cepstral coefficients for robust speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2011, pp. 4784–4787, doi: 10.1109/ICASSP.2011.5947425.

[26] R. W. Floyd and L. Steinberg, "An adaptive algorithm for spatial gray scale," *Proc. Soc. Inf. Display*, vol. 17, pp. 75–77, 1975.

[27] M. Ravanelli and Y. Bengio, "Speaker recognition from raw waveform with sincNet," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2018, pp. 1021–1028, doi: 10.1109/SLT.2018.8639585.

[28] J.-W. Jung, H.-S. Heo, I.-H. Yang, H.-J. Shim, and H.-J. Yu, "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.*, 2018, pp. 5349–5353, doi: 10.1109/ICASSP.2018.8462575.

[29] H. Muckenhirn, M. M. Doss, and S. Marcell, "Towards directly modeling raw speech signal for speaker verification using CNNS," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 4884–4888, doi: 10.1109/ICASSP.2018.8462165.

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