SCALABLE NEURAL ARCHITECTURES FOR END-TO-END ENVIRONMENTAL SOUND CLASSIFICATION

Francesco Paissan, Alberto Ancilotto*, Alessio Brutti, Elisabetta Farella*

Digital Society (DiGis) center - Fondazione Bruno Kessler

ABSTRACT

Sound Event Detection (SED) is a complex task simulating human ability to recognize what is happening in the surrounding from auditory signals only. This technology is a crucial asset in many applications such as smart cities. Here, urban sounds can be detected and processed by embedded devices in an Internet of Things (IoT) to identify meaningful events for municipalities or law enforcement. However, while current deep learning techniques for SED are effective, they are also resource- and power-hungry, thus not appropriate for pervasive battery-powered devices. In this paper, we propose novel neural architectures based on PhiNets for real-time acoustic event detection on microcontroller units. The proposed models are easily scalable to fit the hardware requirements and can operate both on spectrograms and waveforms. In particular, our architectures achieve state-of-the-art performance on UrbanSound8K in spectrogram classification (around 77%) with extreme compression factors (99.8%) with respect to current state-of-the-art architectures.

Index Terms— sound event detection, tinyML, scalable backbone, IoT;

1. INTRODUCTION

The task of Sound Event Detection (SED) consists in recognizing acoustic events in audio streams. This task is of interest both for industrial and smart cities applications. Although recently the research community has steadily improved the effectiveness and accuracy of SED solutions, current neural networks-based approaches are highly demanding in terms of memory footprint and computational complexity. As a consequence, these systems are not suitable for applications requiring pervasive low-power low-cost sensors. In addition, the high computational complexity affects the lifetime of the devices and results in increased energy absorption and related carbon emissions. Nonetheless, it has already been shown [1] how such architectures are not strictly necessary and can be compressed using, for example, knowledge distillation [2] and network pruning [3]. Moreover, when bringing these approaches to edge devices, it is essential to address the variability in computational resources existing between different platforms. Current compression approaches are generally tailored to a specific hardware platform; thus, they require an expensive process to adapt the neural network to new application scenarios. Conversely, neural architectures should scale efficiently to exploit the available computational resources under different operational constraints.

To address the issues above, in this paper we employ the *PhiNets* [4] architecture's family for the first time in an audio task, showing that these models are good candidates for deep-learning-based multimedia analytics at the edge. In addition, we propose a novel scalable backbone, which resembles the scalability principles of the *PhiNets*, while compressing audio models (down to 766 parameters)[1]. Excluding AudioCLIP [5], which however can not run in real-time on any embedded device (including edge GPUs), the proposed models achieve state-of-the-art performance on the UrbanSound8K benchmark [6], using only a fraction of the parameters and of the computations of current architectures (up to a 99.8% reduction in parameters and operations with respect to similarly performing models).

2. RELATED WORKS

In literature, many architectures for SED exist, recently driven by the DCASE series [7]. This section will review the state-of-the-art techniques grouped by input type (spectrogram or waveform) and target platform.

2.1. SED using spectrograms

The most common approach in detecting acoustic events is spectrogram classification. In this paradigm, the waveform is converted into a spectrogram, using stacked Fourier transforms [8], which are then processed by convolutional neural networks (CNNs). Among the approaches in literature, the best performing are VGGish [9], PiczakCNN [10], and SB-CNN [11]. For the sake of this study, we only consider the aforementioned architectures in terms of their classification accuracy on the Urbansound8K benchmark [6] and in terms of their computational requirements. In particular, these three architectures have different structures, but share a high parameter count, with the smallest one being SB-CNN that counts 241 k parameters. SB-CNN was presented in [11] alongside data augmentation techniques, which proved to be the most effective technique to train networks on the UrbanSound8K benchmark, given the small size of the dataset. Overall, these architectures are constituted by a massive number of parameters that could not fit on off-the-shelf MCUs.

2.2. SED using waveform

An emerging trend in audio classification is exploiting one dimensional convolutions (1DConvs) directly on the waveforms. In this case, neural networks learn the filters to be applied to the input signal directly. Many approaches that inject previous knowledge in the filter shape are proposed to ease the training. In SincNet [12], for example, the filters of the first convolution are forced to be band-pass filters. In [13] instead, the proposed architecture exploits the Gammatone filter initialization to boost the classification performance. In ENVNET-V2 [14], the authors use 1DConvs and then exploit bidimensional convolutions on the feature map. Despite the good performance, this comes with a high cost in computational requirements

*These authors contributed equally to this research. This work was partially funded by the EU H2020 project MARVEL (project number: 957337).

[1] Code available at `https://github.com/fpaissan/phinet_pl`

978-1-6654-0540-9/22/$31.00 ©2022 IEEE

(more than 1 M parameters). AudioCLIP [5] learns a bi-dimensional representation of the waveform with a custom backbone [15] and is currently the best performing model on the UrbanSound8k benchmark. However, this high accuracy is achieved by employing extremely large architectures, counting up to 30 M parameters only for the feature extraction. In Wavelet Networks [16], instead, the architecture resembles the wavelet transform to maximize the sound event detection performance by reducing the impact of phase shifts in the signal. Overall, models working on the waveform are less accurate in classifying acoustic events, mainly due to the higher variability of the signals in the time domain. On the positive side, the 1DConv based models have a higher parameter efficiency given that they need to run in only one dimension.

2.3. SED at the edge

Audio processing at the edge (i.e. on embedded platforms) is relevant for both research and industrial applications. Many approaches targeting embedded platforms are already available in the literature for a variety of audio tasks, namely keyword spotting (KWS) [17, 18] and SED [1, 19]. In [1], a student-teacher approach is presented for model compression via knowledge distillation based on joint alignment of the latent representations and cost function optimization for classification. This approach shows promising results in compressing architectures. However, there is an implicit upper bound to the network's performance since it is empirically shown that the performance of the student network will not surpass that of the teacher. [19] proposes a novel architecture where a dilated convolution replaces the recurrent unit. Moreover, the implementation exploits depth-wise separable convolutions [20], which are well-known for their parameter efficiency. Despite this, the parameter count of the architecture is still higher than what could fit on an MCU. Network quantization offers another popular approach [21]. The most computationally efficient uses Binary Neural Networks (BNNs) [22] but compromises classification performance.

In our work, we present two efficient architectures whose lite computational complexity allows their implementation on extremely resource-constrained platforms, like MCUs. In addition, the proposed models are trained from scratch and, thus, are not limited by the knowledge distillation process.

3. HARDWARE-AWARE SCALING WITH PHINETS

When bringing neural architectures on MCUs, one of the most efficient approaches is *hardware-aware* scaling [4]. Using this paradigm, it is possible to optimize the neural architectures to fit on embedded platforms with a negligible drop in performance. However, in order to exploit this scaling principle, we need to avoid an exponential decay of the performance with respect to computational requirements, as often occurs [23]. For these reasons, we present two different architectures (*PhiNets 1D* and *PhiNets 2D*) that are in line with the *hardware-aware* scaling paradigm and work on two different multiply–accumulate (MAC) and memory ranges. We exploit the scalability principles of *PhiNets* [4], a scalable backbone based on a sequence of inverted residual blocks (depicted in Fig. 2), where the shape of each block depends on three hyper-parameters α, β, t_0 that control disjointly the MAC count and memory requirements (FLASH and RAM), respectively - as described in [4].

3.1. *PhiNets* on spectrograms

We introduced some modifications to the original *PhiNets* architecture to tailor it to the SED task and improve the classification performance. In particular, we propose down-sampling the feature map using max-pooling instead of strided convolutions. We also replace the original input block with a standard 2D convolution.
Max-pooling improves the network's overall performance by around 5% on the UrbanSound8K dataset, and changing the input block showed a similar trend. We observed that networks under 2 k parameters and 5 M MAC perform better using a strided convolution for downsampling and a depth-wise separable input block. Despite this small change, the computational load of the *PhiNets* architectures does not change and is analytically described in [4].

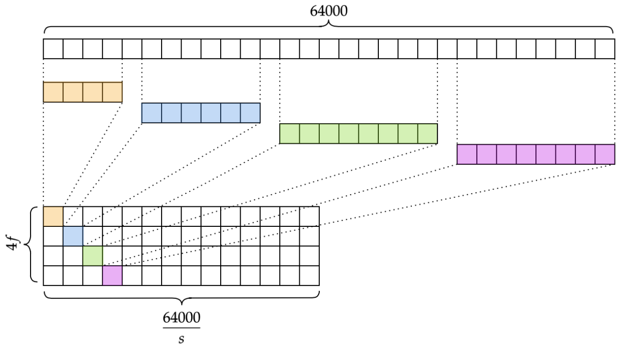

Fig. 1. Illustration of the input 1D convolutional block. The stacked convolutions work on the features, which are time-stretched with different phases. In the illustration, f represents the number of filters, while s represents the stride of the convolution.

3.2. *PhiNets* on waveform

To further reduce the computational cost of the *PhiNets*, we propose a variation of the architecture that works on waveforms, thus exploiting one-dimensional convolutions (1DConvs). By doing this, the relationship between the computational requirements and the number of filters used in each convolutional block is linear instead of quadratic. Moreover, the overall parameter count is lower.

The network architecture is split into three main blocks. The first block is a convolutional block that aims at reducing the shape of the input tensor allowing for a trade-off between accuracy and MAC count. This block consists of four vertically stacked 1DConvs with different kernel sizes (namely 32, 64, 128, 256 points), which work on time-stretched versions of the waveform to avoid losing information in the striding process, as described in Fig. 1. This first convolutional block is followed by a sequence of convolutional blocks composed by a point-wise convolution to up-sample the features, a depth-wise convolution, a squeeze-and-excite block and another point-wise convolution to restore the same number of features as the input. At the end of the network, a fully-connected layer for classification compresses the extracted features and outputs the logits for each class. To decrease the computational complexity and to help the convergence of the network [24], we exploit skip connections in the convolutional blocks. In particular, we either (i) concatenate the input and output tensors to double the number of features used in the following layers (instead of increasing channels by means of e.g. a point-wise convolution) or (ii) sum the input and output tensors.

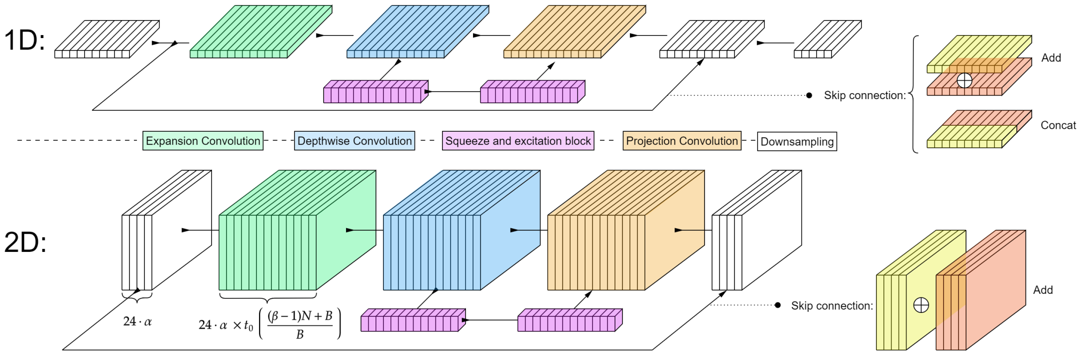

Fig. 2. Illustration of the 1D and 2D convolutional blocks. The input map is fed into the expansion convolution, which affects only the number of channels in the feature map. The feature map is fed into a depth-wise convolution followed by a squeeze-and-excite block. The output of the squeeze-and-excite is projected in a lower dimensionality space via a bottleneck layer. At the end of the 1D block there is an optional downsampling - implemented using average pooling - and a skip connection - either `ADD` or `CONCAT` depending if the layer realizes a downsampling operation or not. For the 2D block, the skip connection always uses an `ADD` operation. In the illustration, B is the number of blocks and N is the ID of the current block.

Fig. 2 shows the convolutional block of the *PhiNets* and how it is performed both in one and two dimensions.

To scale the computational requirements of *PhiNets 1D*, we can change: the number of convolutional blocks, the depth-multiplier, the number of filters in the first convolution or the stride of the input convolution. In particular, changing all of the above parameters has a linear impact on both computational cost (MAC count and parameter count) except for the number of filters in the first convolutional block. The latter has a quadratic impact on both parameter and MAC count. Such scalability features allow for extreme model compression and optimization, while decoupling parameter count and computational cost in alignment with the *harware-aware* scaling paradigm.

4. EXPERIMENTAL SETUP

We benchmarked the two proposed architectures, for waveforms and spectrograms, on the UrbanSound8K dataset [6]. The dataset consists of a collection of 8732 samples of 4 second long typical urban sound events, equally distributed among ten different classes (air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot, jackhammer, siren, and street music). The sampling rate of the original audio sample varies, so we re-sampled each event at 16 kHz, resulting in 64 000 timepoints per sample. We used the standard 10-fold benchmarking procedure for this dataset by averaging the test score after training on eight folds and using one for validation. We augmented the dataset with pitch shift, time-stretching, and Gaussian noise. The model input consists of 40 mel-spectrograms computed on the 4 s sample using 2048 sample windows with a hop-length of 512, resulting in 120 frames for each sound event. For the waveform model instead, we used the re-sampled signal, thus leading to a 64 000 entries input vector.

We trained the models on spectrograms for 100 epochs, with a 10^{-3} learning rate, 10^{-2} weight decay and 0.05 dropout rate in the convolutional blocks. Moreover, we also added label smoothing to help the network avoid over-fitting. For the waveform model, we decreased the learning rate starting from 6×10^{-4} every time the validation accuracy was not improving for 15 consecutive epochs. Moreover, we used L2 regularization as for the other approach.

To demonstrate that scaling *PhiNets* has a marginal impact on the classification accuracy with respect to the compression factor, we benchmarked models in the 0.1-20 MMAC range and with 0.7-30 thousand parameters, which is a typical range for real-time operation with off-the-shelf MCUs. For reproducibility, the generated models are enumerated in Table 1.

5. RESULTS

Table 2 reports the performance achieved by the proposed models against a set of state-of-the-art solutions described in Sec. 2.1. We consider different configurations of our PhiNets implementation, which results in different parameter counts as shown in Table 1. The best performing model in spectrogram classification ([14]) has 101 M parameters and achieves 78% classification accuracy. *PhiNets*, instead, achieve a 76.3% accuracy with only 27 k parameters (i.e. 99.8% compression factor). This result pushes the state-of-the-art in SED on tiny architectures with a higher performance-compression ratio. Note that if we do not consider AudioCLIP [5], for the reasons already discussed, *PhiNets* have competitive results also in waveform classification, delivering an overall drop of around 15% in 10-fold accuracy while using only 2% of the best performing model's parameters.

5.1. Impact of scaling on classification accuracy

From the results reported above, it is clear that the two architectures cover well the operating range of MCUs. In fact, when the performance of the model that works on spectrograms starts decreasing drastically, the one-dimensional model helps extend the operative range while keeping the performance a bit higher, as depicted in Fig. 3. Encoding the information presented in the spectrograms is a much more computationally intensive task with respect to waveform analysis. However, this usually comes with an improvement in classification accuracy since spectrograms are less sensitive to noise and phase shift. Also, by scaling the *PhiNets* on spectrograms to a lower MAC and parameter count, we see that the performance is worse than the one of *PhiNets* when working on raw waveforms. In fact, the two models complement each other when scaling computational requirements, as shown in Fig. 3: spectrum yields the best complexity performance trade-off for high-end platforms whereas

Input	Model name	Input conv type	Max pooling	α	B	t_0	**MMAC**	**Parameters (k)**
Spectrogram	PhiNets M_{40}	Conv2D	True	0.5	3	4.0	43.00	27.1
	PhiNets M_{15}	SeparableConv2D	True	0.5	2	5.0	14.43	32.3
	PhiNets M_5	Conv2D	True	0.2	2	2.0	4.72	3.80
	PhiNets M_3	Conv2D	True	0.1	2	4.0	2.71	2.18
	PhiNets $M_{1.5}$	SeparableConv2D	True	0.1	2	2.0	1.59	2.00
		Input conv stride	Input conv filters	n	d		**MMAC**	**Parameters (k)**
Waveform	PhiNets 1D M_1	300	3	4	4.5		1.34	11.50
	PhiNets 1D $M_{0.5}$	500	2	4	4.5		0.40	5.91
	PhiNets 1D $M_{0.2}$	1600	2	3	2.5		0.06	2.11
	PhiNets 1D $M_{0.1}$	300	1	4	1.5		0.15	1.15
	PhiNets 1D $M_{0.07}$	500	1	3	1.5		0.07	0.766

Table 1. Parameters for generating the architectures presented in this paper. For the spectrogram classification model, the notation is the same as in [4]. For the waveform model, d refers to the depth multiplier while n refers to the number of blocks.

Input	Model	Params (K)	10-fold acc
Spectrogram	PICZAKCNN [10]	26 000	73.7
	SB-CNN [11]	241	73.11
	VGG [9]	77 000	70.74
	Cerutti M_{20k} [25]	30	69
	Cerutti M_{200k} [25]	200	72
	Cerutti M_{2M} [25]	2 000	75
	Cerutti M_{20M} [25]	70 000	76
	PhiNets M40	27.1	**76.3** ± 5.6
	PhiNets M15	32.2	76.1± 5.0
	PhiNets M5	3.80	68.8± 3.1
	PhiNets M3	2.18	65.3± 1.6
	PhiNets M1.5	2.00	62.3± 3.9

Input	Model	Params (K)	10-fold acc
Waveform	AudioCLIP [5]	>30 000	90.01
	ENVNET-V2 [14]	101 000	78
	W11-NET-WL [16]	1 806	68.47± 4.914
	W18-NET-WL [16]	3 759	65.01 ± 5.431
	W34-NET-WL [16]	4 021	66.77± 4.771
	1DCNN [13]	453	62± 6.791
	W-1DCNN-WL [16]	458	62.64± 4.979
	PhiNets 1D M1	**11.5**	59.3± 3.7
	PhiNets 1D M0.5	5.91	56.4± 6.4
	PhiNets 1D M0.2	2.11	48.4± 2.5
	PhiNets 1D M0.1	1.15	46.3± 4.2
	PhiNets 1D M0.07	0.766	43.3± 2.6

Table 2. Comparison between *PhiNets* and other state-of-the-art architectures, considering spectrograms and waveforms as input features. The central column reports the parameter count. The 10-fold accuracy is taken from the original papers. When standard deviation is not available in the paper, it is not reported in the Table. The notation for the models is taken from the original papers. In particular, M_{20k} in [25] refers to the order of magnitude of the parameter count.

using waveforms as input works better if computational constraints are more stringent.

As a rule of thumb, we saw that to maximize performance, the best combination of parameters involved the use of pooling layers, input block composed of a 2D convolution and a base expansion factor between 4 and 6. Smaller models, instead, performed better with a depth-wise separable input block as described in the original paper [4], strided convolutions for downsampling and a t_0 between 2 and 3. Instead, for the waveform model the test accuracy increases linearly with the depth-multiplier and decreases linearly with stride. Therefore, as expected, the best-performing models have a high number of blocks and a low stride; thus, they compress less the information in the input waveform.

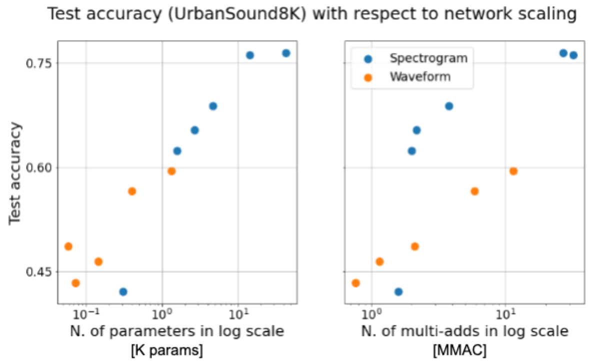

Fig. 3. Classification performance with respect to computational requirements and input type.

6. CONCLUSION AND FUTURE WORK

In this paper, we presented two novel architectures for SED at the edge. Our architectures are the most efficient in computational complexity for spectrogram classification without compromising the classification accuracy. In fact, the best performing model, which is also the biggest we benchmarked in the study, has only 27 k parameters and 43 MMAC and thus can easily fit in an MCU. Moreover, we studied the scalability features of our models in order to validate which architectures are the best performing ones for varying computational requirements. We highlight that *hardware-aware* scaling is the most computationally efficient way of bringing neural networks on MCUs (i.e., extremely low-resource devices). We plan to extend these architectures to other audio tasks, namely keyword spotting and speech recognition and to other training paradigms (e.g. cross-modal, knowledge distillation). Moreover, we will expand the 1DConv model with different input convolutional layers shapes (e.g., Sinc, Wavelet) to boost the models' performance.

7. REFERENCES

[1] Gianmarco Cerutti, Rahul Prasad, Alessio Brutti, and Elisabetta Farella, "Compact recurrent neural networks for acoustic event detection on low-energy low-complexity platforms," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 4, pp. 654–664, 2020.

[2] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[3] Lorenzo Valerio, Franco Maria Nardini, Andrea Passarella, and Raffaele Perego, "Dynamic hard pruning of neural networks at the edge of the internet," *arXiv preprint arXiv:2011.08545*, 2020.

[4] Francesco Paissan, Alberto Ancilotto, and Elisabetta Farella, "PhiNets: a scalable backbone for low-power AI at the edge," *arXiv preprint arXiv:2110.00337*, 2021.

[5] Andrey Guzhov, Federico Raue, Jörn Hees, and Andreas Dengel, "Audioclip: Extending clip to image, text and audio," *arXiv preprint arXiv:2106.13043*, 2021.

[6] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *22nd ACM International Conference on Multimedia (ACM-MM'14)*, Orlando, FL, USA, Nov. 2014, pp. 1041–1044.

[7] Toni Heittola, Annamaria Mesaros, and Tuomas Virtanen, "Acoustic scene classification in dcase 2020 challenge: generalization across devices and low complexity solutions," *arXiv preprint arXiv:2005.14623*, 2020.

[8] Ronald Newbold Bracewell and Ronald N Bracewell, *The Fourier transform and its applications*, vol. 31999, McGraw-Hill New York, 1986.

[9] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al., "Cnn architectures for large-scale audio classification," in *2017 ieee international conference on acoustics, speech and signal processing (icassp)*. IEEE, 2017, pp. 131–135.

[10] Karol J. Piczak, "Environmental sound classification with convolutional neural networks," *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6, 2015.

[11] Justin Salamon and Juan Pablo Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 279–283, 2017.

[12] Mirco Ravanelli and Yoshua Bengio, "Speaker recognition from raw waveform with sincnet," in *2018 IEEE Spoken Language Technology Workshop (SLT)*, 2018, pp. 1021–1028.

[13] Pablo Zinemanas, Pabo Cancela, and Martín Rocamora, "End-to-end convolutional neural networks for sound event detection in urban environments," 04 2019.

[14] Yuji Tokozume and Tatsuya Harada, "Learning environmental sounds with end-to-end convolutional neural network," *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2721–2725, 2017.

[15] Andrey Guzhov, Federico Raue, Jörn Hees, and Andreas Dengel, "Esresne (x) t-fbsp: Learning robust time-frequency transformation of audio," *arXiv preprint arXiv:2104.11587*, 2021.

[16] David W Romero, Erik J Bekkers, Jakub M Tomczak, and Mark Hoogendoorn, "Wavelet networks: Scale equivariant learning from raw waveforms," *arXiv preprint arXiv:2006.05259*, 2020.

[17] Yundong Zhang, Naveen Suda, Liangzhen Lai, and Vikas Chandra, "Hello edge: Keyword spotting on microcontrollers," *arXiv preprint arXiv:1711.07128*, 2017.

[18] Alice Coucke, Mohammed Chlieh, Thibault Gisselbrecht, David Leroy, Mathieu Poumeyrol, and Thibaut Lavril, "Efficient keyword spotting using dilated convolutions and gating," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6351–6355.

[19] Konstantinos Drossos, Stylianos I Mimilakis, Shayan Gharib, Yanxiong Li, and Tuomas Virtanen, "Sound event detection with depthwise separable and dilated convolutions," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–7.

[20] François Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[21] Jiwei Yang, Xu Shen, Jun Xing, Xinmei Tian, Houqiang Li, Bing Deng, Jianqiang Huang, and Xian-sheng Hua, "Quantization networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7308–7316.

[22] Gianmarco Cerutti, Renzo Andri, Lukas Cavigelli, Elisabetta Farella, Michele Magno, and Luca Benini, "Sound event detection with binary neural networks on tightly power-constrained iot devices," in *Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design*, 2020, pp. 19–24.

[23] Mingxing Tan and Quoc Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 6105–6114.

[24] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein, "Visualizing the loss landscape of neural nets," in *NIPS'18: Proceedings of the 32nd International Conference on Neural Information Processing Systems*. Curran Associates Inc., 2018, pp. 6391–6401.

[25] Gianmarco Cerutti, Rahul Prasad, Alessio Brutti, and Elisabetta Farella, "Neural Network Distillation on IoT Platforms for Sound Event Detection," in *Proc. Interspeech 2019*, 2019, pp. 3609–3613.