



# Global-Local Convolution with Spiking Neural Networks for Energy-efficient Keyword Spotting

Shuai Wang<sup>1</sup>, Dehao Zhang<sup>1</sup>, Kexin Shi<sup>1</sup>, Yuchen Wang<sup>1</sup>, Wenjie Wei<sup>1</sup>, Jibin Wu<sup>2</sup>, Malu Zhang<sup>1,\*</sup>

<sup>1</sup>University of Electronic Science and Technology of China

<sup>2</sup>The Hong Kong Polytechnic University

wangshuai718@std.uestc.edu.cn, jibin.wu@polyu.edu.hk, maluzhang@uestc.edu.cn

## Abstract

Thanks to Deep Neural Networks (DNNs), the accuracy of Keyword Spotting (KWS) has made substantial progress. However, as KWS systems are usually implemented on edge devices, energy efficiency becomes a critical requirement besides performance. Here, we take advantage of spiking neural networks' energy efficiency and propose an end-to-end lightweight KWS model. The model consists of two innovative modules: 1) Global-Local Spiking Convolution (GLSC) module and 2) Bottleneck-PLIF module. Compared to the hand-crafted feature extraction methods, the GLSC module achieves speech feature extraction that is sparser, more energy-efficient, and yields better performance. The Bottleneck-PLIF module further processes the signals from GLSC with the aim to achieve higher accuracy with fewer parameters. Extensive experiments are conducted on the Google Speech Commands Dataset (V1 and V2). The results show our method achieves competitive performance among SNN-based KWS models with fewer parameters.

**Index Terms:** Keyword spotting, Spiking neural networks, Global-Local spiking convolution.

## 1. Introduction

Keyword Spotting (KWS) systems recognize predefined commands, which are always deployed on edge devices as an interface for human-machine interaction. Current mainstream KWS models implemented with Artificial Neural Networks (ANNs) [1] achieve outstanding accuracy. However, limited by endurance, CPU resources, and portability, deploying and running ANNs on edge devices for extended periods can be difficult. Therefore, designing a high-accuracy, lightweight, and energy-efficient KWS model for edge devices is a hot topic that awaits a solution.

As the third-generation neural networks, Spiking Neural Networks (SNNs) [2, 3] have gained widespread attention due to their asynchronous event-driven architecture [4] and ultra-low energy consumption. The spiking event-driven mechanism [5] of SNNs computes only when necessary, resulting in sparse information transmission and significantly reduced energy consumption. This is suitable for resource-constrained edge devices. Besides, it has been proved that accumulate (AC) operation is compact and energy-efficient compared with Multiply-and-Accumulate (MAC) operation [6]. Therefore, when deployed on hardware, SNNs using AC operation expend much less energy in comparison with MAC-dependent DNNs [7].

The advantages of SNNs have motivated many researchers to apply them to KWS tasks [8, 9]. However, many attempts based on deep SNNs still use FFT [10], MFCC [11] to pre-

process raw speech (wav) which needs massive computing resources. This goes against to our original intention of using SNNs to implement energy-efficient KWS models. To avoid this problem, some researchers have attempted to directly utilize raw speech signals with low-resource-consuming convolutional operations [12]. For instance, Philipp et al. [13] proposed the end-to-end streaming model by dilated convolution with  $stride = 1$ . However, this approach fails to compress the length of long speech sequences, leading to evident feature redundancy. What's more, Yang et al. [14] proposed an end-to-end deep residual SNN model, and it demonstrated a notably high level of recognition accuracy. However, their approach employs the integrate-and-fire (IF) neuron model, which lacks a membrane potential decay mechanism. This will result in frequent spikes firing, leading to higher computational energy consumption in SNNs[15].

In this paper, we constructed an end-to-end SNN-based KWS model to address the issues above. We tested the accuracy of our model on the Google Speech Commands datasets [16] (V1 and V2) and compared its model size with the related works based on SNNs. Encouragingly, compared with similar SNN-based models, our model achieves competitive performance on smaller model sizes. Finally, energy efficiency calculations prove that our model consumes  $10\times$  less energy than ANNs with the same structure. Hence, our SNN-KWS model aligns perfectly with the requirements of edge devices for high accuracy, lightweight structure, and low energy consumption. The major contributions of this paper can be summarized as follows:

- Global-Local Spiking Convolution (GLSC) module: We design the GLSC module to achieve better and more energy-efficient spiking convolution. It can compress the length of long speech sequences layer by layer while considering both global and local features.
- Bottleneck-PLIF module: To achieve a more lightweight and efficient SNN architecture, we combine the Bottleneck structure in ResNet [17] with more efficient Parametric Leaky Integrate-and-Fire (PLIF) [15] neurons to create a more lightweight classifier.
- By integrating the proposed GLSC and Bottleneck-PLIF modules, we construct a novel end-to-end SNN-KWS model. Our SNN-KWS model achieves competitive performance in both accuracy and parameter efficiency within the domain of SNN-based models.

## 2. Preliminaries

In this section, we will give an overview of two essential components in our model: end-to-end speech feature extraction and spiking neural networks. Additionally, we will analyze the challenges associated with these components.

\*Corresponding author.

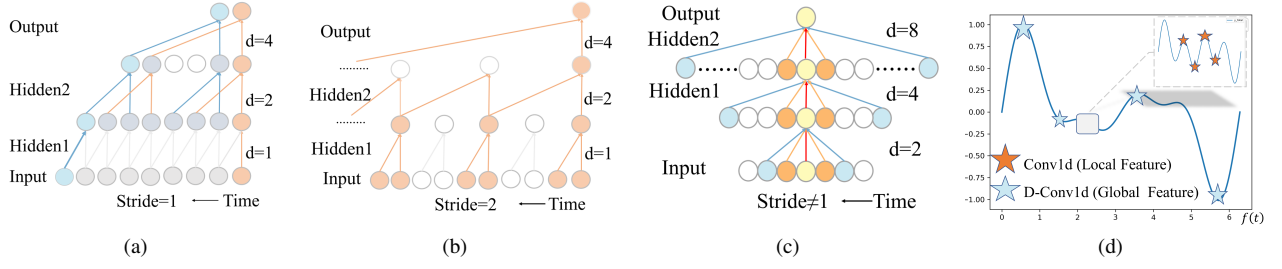


Figure 1: A comparative analysis between single convolution and global-local convolution. (a) Dilated Conv1d with  $\text{stride} = 1$ . The hidden layers and output features are highly redundant, as evidenced by the gray blocks representing the overlapping features. (b) Dilated Conv1d with  $\text{stride} \neq 1$ . The receptive field exponentially increases with the dilation factor  $d$ , leading to a loss of local information as white blocks. (c) and (d) the Global-Local convolution method. It can achieve a good balance between global and local features in long speech sequences, and maintain a consistent focus on local features when the  $\text{stride} \neq 1$ .

### 2.1. End-to-end Convolution for Speech Features

To alleviate the energy consumption caused by conventional speech feature extraction methods, such as FFT [10], MFCC [11], the most popular approach is to use direct convolution methods for end-to-end feature extraction. They can be summarized as:

$$f(t) * g(t) = \int_0^t f(u) g(t-u) du \quad (1)$$

$g(t)$  is regarded as different convolution kernels and  $f(t)$  is denoted as original speech wave sequences. While the convolution methods have been successfully applied in speech feature extraction, certain challenges still require further resolution.

For example, the parameters in Conv1d [18] increase with the expansion of the receptive field, resulting in the redundancy of parameters in  $g(t)$ . The dilated Conv1d (D-Conv1d) [19] was proposed to address this problem. However, as illustrated in Figs 1(a) and 1(b), dilated convolution [13] with  $\text{stride} = 1$  suffers from redundancy of features and  $\text{stride} \neq 1$  may lead to loss of local features. Therefore, there is no convolution method currently that can simultaneously address both the redundancy of parameters and the loss of local features.

### 2.2. Spiking Neural Networks

SNNs encode information through binary spikes over time and work in an event-driven manner, which has a great advantage in energy consumption. As a basic unit of SNNs, various spiking neurons are proposed to emulate the mechanism of biological neurons. Among them, the Leaky Integrate-and-Fire (LIF) [20] model is widely used due to its simplicity. The dynamics of a LIF neuron can be expressed as follows:

$$U_i^{t+1,n} = \tau U_i^{t,n} + \sum_{j=1}^{l(n-1)} w_{ij}^n O_j^{t+1,n-1} \quad (2)$$

where  $\tau$  is the constant leaky factor,  $U_i^{t+1,n}$  is the membrane potential of neuron  $i$  in the  $n$ th layer at the time step  $t+1$ , and  $\sum_{j=1}^{l(n-1)} w_{ij}^n O_j^{t+1,n-1}$  denotes the pre-synaptic inputs for neuron  $i$ . When the membrane potential  $U_i^{t+1,n}$  exceeds the firing threshold  $V_{th}$ , the neuron  $i$  fires a spike  $O_i^{t+1,n}$  and  $U_i^{t+1,n}$  reset to 0. The firing function and hard reset mechanism can be described by Eq. 3 and Eq. 4, respectively.

$$O_i^{t+1,n} = H(U_i^{t+1,n} - V_{th}) \quad (3)$$

$$U_i^{t+1,n} = U_i^{t+1,n}(1 - O_i^{t+1,n}) \quad (4)$$

where  $H$  denotes the Heaviside step function.

Many studies have effectively leveraged the energy efficiency of spiking neurons to develop energy-efficient SNN-KWS models [21, 22]. However, these methods have not taken into account the lightweight structural requirements of edge devices. Therefore, we aim to design a more lightweight and energy-efficient SNN-KWS model by utilizing more advanced structures and spiking neuron models.

## 3. Method

In this section, we propose an end-to-end SNN-KWS model that effectively addresses the limitations mentioned in Section 2. The overall structure of the model is illustrated in Fig.2, which mainly comprises two innovative modules: 1) the GLSC module and 2) the Bottleneck-PLIF module.

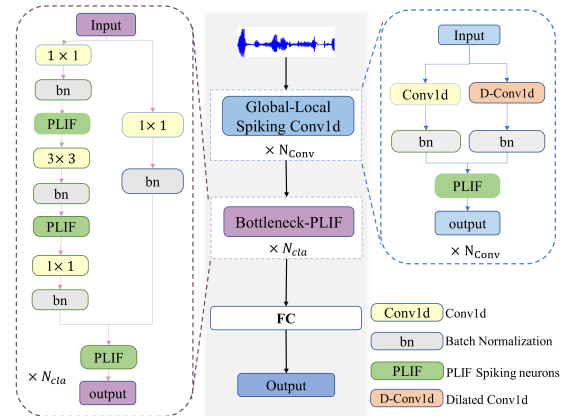


Figure 2: Our SNN-KWS model structure. It consists of  $N_{Conv} = 4$  GLSC blocks (right part) for better feature extraction, and  $N_{Cla} = 2$  Bottleneck-PLIF blocks (left part) for effective classification.

### 3.1. Global-Local Spiking Conv1d Module

To achieve better and energy-efficient speech feature extraction, we propose a Global-Local Spiking Conv1d module for end-to-end feature extraction. GLSC mainly consists of three components, Conv1d, D-Conv1d, and spiking neurons. The flowchart

of the GLSC module is illustrated in the right portion of Fig. 2, and it can be mathematically expressed as:

$$outputs = H(bn(g_1(t) * f(t)) + bn(g_2(t) * f(t))) \quad (5)$$

where  $g_1, g_2$  are the convolution kernels of Conv1d and D-Conv1d, respectively.  $bn$  is Batch Normalization and  $H$  is the firing function of spiking neurons as Eq.3. In the following, we will analyze how the proposed GLSC can achieve enhanced feature extraction and energy efficiency.

In contrast to a single Conv1d or D-Conv1d method, the proposed GLSC module can benefit from both worlds. As demonstrated in Fig. 1(d), the GLSC module can effectively balance local and global features in long speech sequences. Although the idea of global-local feature extraction exists in some ANN-based models such as Branchformer [23], their success relies on utilizing complex attention mechanisms rather than combining global and local convolutions directly. In ANNs, merging two convolutions directly leads to feature disappearance, where salient details like the orange block become indistinct upon addition as shown in Fig.3.

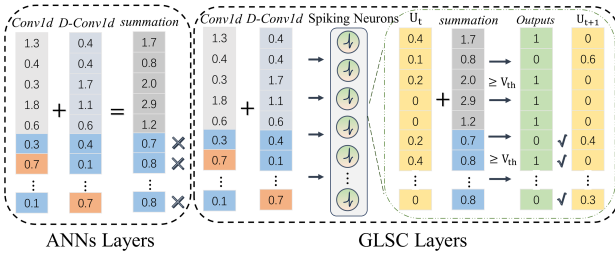


Figure 3: The Global-local convolution feature extraction in ANNs and GLSC layers.  $U_{t+1}$  represents the membrane potential contribution of spiking neurons after decaying from  $U_t$ .

Here, we innovatively employ spiking neurons to solve this problems and achieve simpler and sparser global-local feature extraction. As shown in the right part of Fig.3, the output of spiking neurons at  $t$  depends not only on the summation but also considers the residual membrane potential from  $t-1$ , significantly mitigating feature disappearance caused by the summation. We will validate this aspect through ablation studies. Moreover, only when the  $summation + U_t$  is greater than the  $V_{th}$ , can spiking neurons pass key information for outputs as the green blocks. This approach effectively prevents the accumulation of irrelevant features across layers, while ensuring that the feature vectors become sparser. So the GLSC module achieves a more efficient and sparser end-to-end feature extraction.

### 3.2. Bottleneck-PLIF Module

To address the limitations of the existing SNN-based KWS models, we take advantage of the effective PLIF spiking neuron and lightweight bottleneck structure to construct the Bottleneck-PLIF module.

$$U_i^{t+1,n} = U_i^{t,n} - k(a) \left( U_i^{t,n} - \sum_{j=1}^{l(n-1)} w_{ij}^n O_j^{t+1,n-1} \right) \quad (6)$$

Eq. 6 depicts the membrane potential dynamics of a PLIF neuron. Compared to traditional LIF neurons, the PLIF neuron exhibits two notable enhancements. First, learnable  $k(a)$  replaces the constant decay hyperparameters  $\tau$  in Eq.2, which can be optimized during training. Second, PLIF applies learnable  $k(a)$

to the input. As depicted in Fig.4, neurons exhibit a greater diversity of outputs when subjected to different  $\tau$  under the same input conditions.

Meanwhile, inspired by the Bottleneck block in ResNet [17], it can efficiently integrate feature information with fewer parameters and reduce feature dimensionality by fusing channels. We incorporate PLIF into the Bottleneck structure to achieve a more efficient spiking classifier as shown in the left part of Fig. 2, and its mathematical expression is as follows:

$$Outputs = H[f_1(H(f_3(H(f_1(Input))) + f_1(Input))] \quad (7)$$

$f_n$  represents  $n \times n$  convolution and Batch Normalization, and  $H$  represents the firing function of PLIF. In Eq.7,  $f_1$  are used for fusing channels without compromising the input features, which can reduce feature dimensionality without compromising the original structure.  $f_3$  are used to further compute the previous spikes features, which can further process the signals from the GLSC module with a more lightweight model size.

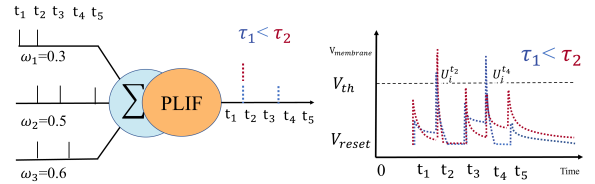


Figure 4: with the same inputs, these neurons with different  $\tau$  result in varied leaky rates for neurons' membrane potential(right part), thereby leading to diverse output results(left part).

## 4. Experiments

### 4.1. Dataset

The Google Speech Commands (GSC) [16] dataset includes 30 short commands for Version 1 (V1) and 35 for Version 2 (V2), recorded by 1,881 and 2,618 speakers, respectively. To make a fair comparison, our experiments are conducted on the 12-class classification and 35-class classification tasks as previous SNN models [14, 24]. While 12-class classification recognizes 12 classes, that include 10 commands: “yes”, “no”, “up”, “down”, “left”, “right”, “on”, “off”, “stop”, “go”, and two additional classes: silence, and an unknown class. The unknown class covers the remaining 20 (25) speech commands in the set of 30 (35). The silence class accounting for about 10 % of the total dataset is generated by splicing the noise files in the dataset. Finally, GSC-V1 is split into 56588 training, 7743 validation, and 7835 test utterances, and GSC-V2 is divided into 92843 training, 11003 validation, and 12005 test utterances. We use the STBP[25] method to train the entire model directly.

### 4.2. Accuracy and Model Size

To validate the accuracy and model size of our proposed model, we conduct a comprehensive comparative analysis with previous studies [26, 14, 13, 24, 21, 22, 27, 28]. The experimental results are shown in Table 1. Although our accuracy is slightly lower compared to ST-Attention-SNN and SRNN+ALIF, our model size is significantly smaller. In conclusion, our KWS-SNN achieves competitive performance in both 12-class and 35-class tasks with a substantially reduced model size. This indicates that our model can be easier to deploy on edge devices.

Table 1: A summary of KWS models' accuracy and model size.

Model	Model Size(K)	Acc(%)
Google Speech Commands Dataset Version 1 (12)		
NLIF full SNN[26]	220	87.9
E2E residual SNN[14]	86.5	92.2
(Our) SNN-KWS	<b>70.1</b>	<b>93.0</b>
Google Speech Commands Dataset Version 2 (12)		
ST-Attention-SNN [21]	2170	95.1
SLAYER-RF-CNN [24]	280	91.4
SpikGRU[29]	111	94.9
(Our) SNN-KWS	<b>70.1</b>	<b>94.4</b>
Google Speech Commands Dataset Version 2 (35)		
WaveSense [13]	N/A	79.5
LSTMs-SNN [22]	N/A	91.5
SRNN+ALIF [27]	222.1	92.5
Speech2Spikes[28]	410	89.5
(Our) SNN-KWS	<b>80.2</b>	<b>92.9</b>

### 4.3. Energy Efficiency

In this part, we validate the energy efficiency advantage of our model over their ANNs counterparts. According to the standards established in the field of neuromorphic computing [30], the energy consumption ratio between our model and an equivalent ANN model can be calculated as:

$$Energy_{rate} = \frac{AC}{MAC} * SpikingRate * TimeSteps \quad (8)$$

$\frac{AC}{MAC}$  is denoted as the energy consumption ratio between float-point additions(AC) in SNNs and float-point multiplications(MAC) in ANNs. Extensive research has substantiated that  $\frac{AC}{MAC} = \frac{1}{7}$  [31]. *SpikingRate* and *TimeSteps* represent the average firing rate and simulation time window. As illustrated in Fig.5, the average spike firing rate of each module is 8.3% and the *TimeSteps* in our model is set to 8. Therefore, according to Eq.8, our SNN-KWS model achieves more than 10× energy saving over the ANNs counterpart.

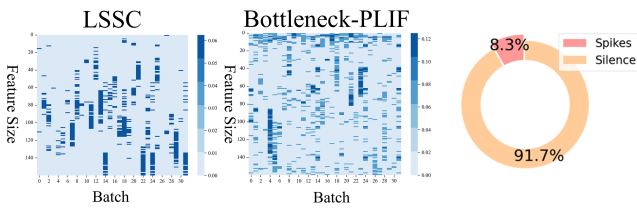


Figure 5: The average spike firing rate of our SNN-KWS model when *TimeSteps* is 8 on the GSC-V1 dataset. The average spike firing rate of the entire network is approximately 8.3%.

### 4.4. Ablation Study

In this part, we conduct ablation studies to validate the effectiveness of the GLSC and Bottleneck-PLIF modules, respectively. Firstly, we evaluate the GLSC by comparing it with single convolution methods on the same number of parameters. As illustrated in Figs.6(a) and 6(b), the GLSC module consistently surpasses other methods (black and green), exhibiting both better

performance and convergence. It is noteworthy that the GLC-ANN(blue curve) represents the substitution of spiking neurons in the GLSC module with continuous activation functions of ANNs. By comparing the red and blue curves, it can be proven that spiking neurons play a key role in addressing the issue of feature disappearance.

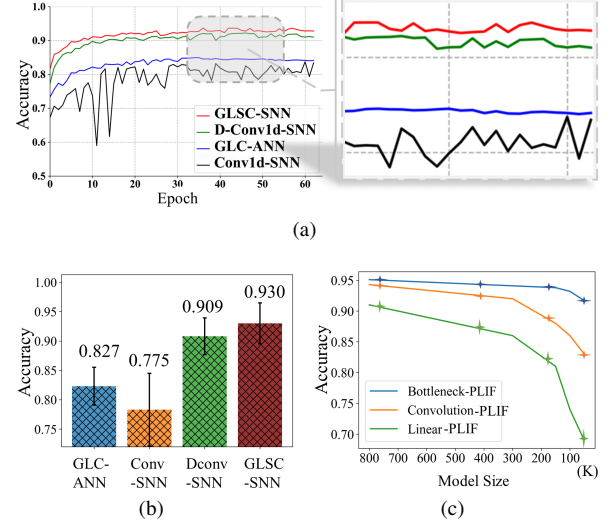


Figure 6: Ablation Studies. (a,b)Validating the feature extraction capabilities of the GLSC module. (c) The performance advantage of the Bottleneck-PLIF module becomes more pronounced as the number of parameters decreases.

Next, we verify that the Bottleneck module can allow us to achieve better performance while utilizing fewer parameters. As shown in Fig.6(c), the performance of all classifiers exhibits a decline as parameters decrease. However, the reduction in parameters has minimal impact on our Bottleneck-PLIF model, and our method can achieve an accuracy of 93% even when the parameters are below 100K.

## 5. Conclusion

In this work, we propose a novel SNN-KWS model with two innovative modules. The GLSC module enhanced end-to-end convolution speech feature extraction. It avoids the high computation costs associated with traditional data pre-processing [10, 11], while simultaneously considering both global and local speech features. The Bottleneck-PLIF module further calculates the spike features from the GLSC module, with the aim of achieving higher classification accuracy using fewer parameters. By conducting experiments on the GSC [16] dataset, our model achieves competitive performance in both accuracy and parameter efficiency among similar SNN-based models and achieves more than 10× energy saving over the ANNs. Therefore, our SNN-KWS model proficiently satisfies the requirements of edge devices in terms of exceptional accuracy, lightweight design, and energy efficiency. In the future, we will implement it realistically on a neuromorphic chip.

## 6. Acknowledgements

This work was supported by the National Science Foundation of China under Grant 62106038, and in part by the Sichuan Science and Technology Program under Grant 2023YFG0259.



## 7. References

- [1] Z. Yang, S. Sun, J. Li, X. Zhang, X. Wang, L. Ma, and L. Xie, "CaTT-KWS: A Multi-stage Customized Keyword Spotting Framework based on Cascaded Transducer-Transformer," in *Proc. Interspeech 2022*, 2022, pp. 1681–1685, doi:10.21437/interspeech.2022-10258.
- [2] R.-J. Zhu, M. Zhang, Q. Zhao, H. Deng, Y. Duan, and L.-J. Deng, "Tcja-snn: Temporal-channel joint attention for spiking neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2024.
- [3] M. Zhang, J. Wang, J. Wu, A. Belatreche, B. Amornpaisannon, Z. Zhang, V. P. K. Miriyala, H. Qu, Y. Chua, T. E. Carlson *et al.*, "Rectified linear postsynaptic potential function for backpropagation in deep spiking neural networks," *IEEE transactions on neural networks and learning systems*, vol. 33, no. 5, pp. 1947–1958, 2021.
- [4] H. Akolkar, C. Meyer, X. Clady, O. Marre, C. Bartolozzi, S. Panzeri, and R. Benosman, "What can neuromorphic event-driven precise timing add to spike-based pattern recognition?" *Neural computation*, vol. 27, no. 3, pp. 561–593, 2015, doi:10.1162/neco.2013.0703.
- [5] W. Wei, M. Zhang, J. Zhang, A. Belatreche, J. Wu, Z. Xu, X. Qiu, H. Chen, Y. Yang, and H. Li, "Event-driven learning for spiking neural networks," *arXiv preprint arXiv:2403.00270*, 2024.
- [6] R. Karmakar, S. Chattopadhyay, and S. Chakraborty, "Impact of ieee 802.11 n/ac phy/mac high throughput enhancements on transport and application protocols—a survey," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2050–2091, 2017, doi:10.14456/easr.2021.60.
- [7] W. Maass, "Networks of spiking neurons: the third generation of neural network models," *Neural networks*, vol. 10, no. 9, pp. 1659–1671, 1997, doi:10.1016/s0893-6080(97)00011-7.
- [8] Z. Pan, Y. Chua, J. Wu, M. Zhang, H. Li, and E. Ambikairajah, "An efficient and perceptually motivated auditory neural encoding and decoding algorithm for spiking neural networks," *Frontiers in neuroscience*, vol. 13, p. 1420, 2020, doi:10.3389/fnins.2019.01420.
- [9] J. Wu, Y. Chua, M. Zhang, H. Li, and K. C. Tan, "A spiking neural network framework for robust sound classification," *Frontiers in neuroscience*, vol. 12, p. 836, 2018, doi:10.3389/fnins.2018.00836.
- [10] K. Kim, C. Gao, R. Graça, I. Kiselev, H.-J. Yoo, T. Delbruck, and S.-C. Liu, "A 23 $\mu$ w solar-powered keyword-spotting asic with ring-oscillator-based time-domain feature extraction," in *2022 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 65. IEEE, 2022, pp. 1–3, doi:10.1109/isscc42614.2022.9731708.
- [11] V. Tiwari, "Mfcc and its applications in speaker recognition," *International journal on emerging technologies*, vol. 1, no. 1, pp. 19–22, 2010.
- [12] S. Phiphitphatphaisit and O. Surinta, "Deep feature extraction technique based on conv1d and lstm network for food image recognition," *Engineering and Applied Science Research*, vol. 48, no. 5, pp. 581–592, 2021, doi:10.1109/comst.2017.2745052.
- [13] P. Weidel and S. Sheik, "Wavesense: Efficient temporal convolutions with spiking neural networks for keyword spotting," *arXiv preprint arXiv:2111.01456*, 2021, doi:10.48550/arXiv.2111.01456.
- [14] Q. Yang, Q. Liu, and H. Li, "Deep residual spiking neural network for keyword spotting in low-resource settings," *Proc. Interspeech 2022*, pp. 3023–3027, 2022, doi:10.21437/interspeech.2022-107.
- [15] W. Fang, Z. Yu, Y. Chen, T. Masquelier, T. Huang, and Y. Tian, "Incorporating learnable membrane time constant to enhance learning of spiking neural networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2661–2671, doi:10.1109/iccv48922.2021.00266.
- [16] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *arXiv preprint arXiv:1804.03209*, 2018, doi:10.48550/arXiv.1804.03209.
- [17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, doi:10.1109/cvpr.2016.90.
- [18] R. Johnson and T. Zhang, "Deep pyramid convolutional neural networks for text categorization," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 562–570, doi:10.18653/v1/p17-1052.
- [19] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *4th International Conference on Learning Representations, ICLR*, Y. Bengio and Y. LeCun, Eds., 2016, doi:abs/1511.07122.
- [20] Y. Wu, L. Deng, G. Li, J. Zhu, Y. Xie, and L. Shi, "Direct training for spiking neural networks: Faster, larger, better," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 1311–1318, doi:10.1609/aaai.v33i01.33011311.
- [21] Y. Wang, K. Shi, C. Lu, Y. Liu, M. Zhang, and H. Qu, "Spatial-temporal self-attention for asynchronous spiking neural networks," in *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, Edith Elkind, Ed., vol. 8, 2023, pp. 3085–3093, doi:10.24963/ijcai.2023/344.
- [22] S. Zhang, Q. Yang, C. Ma, J. Wu, H. Li, and K. C. Tan, "Long short-term memory with two-compartment spiking neuron," *arXiv preprint arXiv:2307.07231*, 2023, doi:10.48550/arXiv.2307.07231.
- [23] Y. Peng, S. Dalmia, I. Lane, and S. Watanabe, "Branchformer: Parallel mlp-attention architectures to capture local and global context for speech recognition and understanding," in *International Conference on Machine Learning*. PMLR, 2022, pp. 17 627–17 643, doi:10.26434/chemrxiv-2022-07231.
- [24] G. Orchard, E. P. Frady, D. B. D. Rubin, S. Sanborn, S. B. Shrestha, F. T. Sommer, and M. Davies, "Efficient neuromorphic signal processing with loihi 2," in *2021 IEEE Workshop on Signal Processing Systems (SiPS)*. IEEE, 2021, pp. 254–259, doi:10.1109/sips52927.2021.00053.
- [25] Y. Wu, L. Deng, G. Li, J. Zhu, and L. Shi, "Spatio-temporal back-propagation for training high-performance spiking neural networks," *Frontiers in neuroscience*, vol. 12, p. 331, 2018, doi:doi.org/10.3389/fnins.2018.00331.
- [26] T. Pellegrini, R. Zimmer, and T. Masquelier, "Low-activity supervised convolutional spiking neural networks applied to speech commands recognition," in *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2021, pp. 97–103, doi:10.1109/slt48900.2021.9383587.
- [27] B. Yin, F. Corradi, and S. M. Bohtë, "Accurate and efficient time-domain classification with adaptive spiking recurrent neural networks," *Nature Machine Intelligence*, vol. 3, no. 10, pp. 905–913, 2021, doi:10.1101/2021.03.22.436372.
- [28] K. M. Stewart, T. Shea, N. Pacik-Nelson, E. Gallo, and A. Danielescu, "Speech2spikes: Efficient audio encoding pipeline for real-time neuromorphic systems," in *Proceedings of the 2023 Annual Neuro-Inspired Computational Elements Conference*, 2023, pp. 71–78, doi:10.1145/3584954.3584995.
- [29] M. Dampfhofer, T. Mesquida, E. Hardy, A. Valentian, and L. Anghel, "Leveraging sparsity with spiking recurrent neural networks for energy-efficient keyword spotting," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [30] A. Sengupta, Y. Ye, R. Wang, C. Liu, and K. Roy, "Going deeper in spiking neural networks: Vgg and residual architectures," *Frontiers in neuroscience*, vol. 13, p. 95, 2019, doi:10.3389/fnins.2019.00095.
- [31] M. Horowitz, "1.1 computing's energy problem (and what we can do about it)," in *2014 IEEE international solid-state circuits conference digest of technical papers (ISSCC)*. IEEE, 2014, pp. 10–14, doi:10.1109/isscc.2014.6757323.