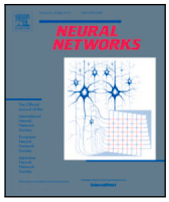
Neural Networks 176 (2024) 106330

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet



Full Length Article

Multi-scale full spike pattern for semantic segmentation

Qiaoyi Su [a,c], Weihua He [b], Xiaobao Wei [c], Bo Xu [a,c], Guoqi Li [a,d,*]

[a] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China
[b] Department of Precision Instrument, Tsinghua University, Beijing 100084, China
[c] Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[d] Institute of Automation, Key Laboratory of Brain Cognition and Brain-inspired Intelligence Technology, Chinese Academy of Sciences, Beijing 100190, China

ARTICLE INFO

Dataset link: https://github.com/BICLab/MFS-Seg

Keywords:
Spiking neural network
Semantic segmentation
Neuromorphic computing
Deep neural network
Energy efficiency
Brain-inspired computing

ABSTRACT

Spiking neural networks (SNNs), as the brain-inspired neural networks, encode information in spatio-temporal dynamics. They have the potential to serve as low-power alternatives to artificial neural networks (ANNs) due to their sparse and event-driven nature. However, existing SNN-based models for pixel-level semantic segmentation tasks suffer from poor performance and high memory overhead, failing to fully exploit the computational effectiveness and efficiency of SNNs. To address these challenges, we propose the multi-scale and full spike segmentation network (MFS-Seg), which is based on the deep direct trained SNN and represents the first attempt to train a deep SNN with surrogate gradients for semantic segmentation. Specifically, we design an efficient fully-spike residual block (EFS-Res) to alleviate representation issues caused by spiking noise on different channels. EFS-Res utilizes depthwise separable convolution to improve the distributions of spiking feature maps. The visualization shows that our model can effectively extract the edge features of segmented objects. Furthermore, it can significantly reduce the memory overhead and energy consumption of the network. In addition, we theoretically analyze and prove that EFS-Res can avoid the degradation problem based on block dynamical isometry theory. Experimental results on the Camvid dataset, the DDD17 dataset, and the DSEC-Semantic dataset show that our model achieves comparable performance to the mainstream UNet network with up to 31× fewer parameters, while significantly reducing power consumption by over 13×. Overall, our MFS-Seg model demonstrates promising results in terms of performance, memory efficiency, and energy consumption, showcasing the potential of deep SNNs for semantic segmentation tasks. Our code is available in https://github.com/BICLab/MFS-Seg.

1. Introduction

Semantic segmentation, as a fundamental task in the computer vision field for scene understanding, aims at classifying each pixel and labeling it according to its category. Most previous artificial neural networks (ANNs) focus on improving the performance of the model at the expense of computational efficiency (Soylu et al., 2023; Kirillov et al., 2023). When applied in automotive systems, the Internet of Thing (IoT) devices, wearable devices and so on, it is essential to ensure the high performance of the model while being memory and energy efficient. Spiking neural networks, being biologically plausible and energy efficient (Li et al., 2023; Maass, 1997), are potentially applicable as an implementation for computationally efficient segmentation task. Different from traditional ANNs that transmit signals with continuous values, SNNs propagate binary signals (spikes) among neurons, which reduces data transmission and storage overhead. Moreover, SNNs possess asynchronous computation and event-driven properties, when deployed to neuromorphic chips such as TrueNorth (Merolla et al., 2014), Loihi (Davies et al., 2018), and Tianjic (Pei, Deng, et al., 2019), SNNs enable energy reductions up to 1,000 times compared to ANNs.

Currently SNNs for image segmentation include two main training methods: ANN-to-SNN conversion and directly trained SNNs. The former is limited by the accuracy of the original ANN model, while requiring hundreds or even thousands of time steps (Li, He, Dong, Kong, & Zeng, 2022; Patel, Hunsberger, Batir, & Eliasmith, 2021), which implies that real-time inference is extremely challenging. Additionally, the converted SNN methods are not suitable for the sparse event data since their dynamics are designed to approximate the expected activation of the ANN and cannot represent the spatio-temporal information of the DVS data (Deng et al., 2020). The latter approach utilizes surrogate gradient to directly train SNNs, which can reach high performance in a very short time step. However, the existing network structures, such as Spiking-DeepLab and Spiking-FCN (Kim, Chough, & Panda, 2022), are extremely shallow, which is not sufficiently to invoke the effectiveness

* Corresponding author at: Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China.
E-mail addresses: suqiaoyi2020@ia.ac.cn (Q. Su), hwh20@mails.tsinghua.edu.cn (W. He), guoqi.li@ia.ac.cn (G. Li).

https://doi.org/10.1016/j.neunet.2024.106330
Received 17 August 2023; Received in revised form 8 February 2024; Accepted 19 April 2024
Available online 20 April 2024
0893-6080/© 2024 Published by Elsevier Ltd.

of the directly trained SNNs. Overlaying network layers directly on these direct-connected structures would not improve performance and even yield spike degradation problems (Zheng, Wu, Deng, Hu, & Li, 2021).

Therefore, the main challenge for semantic segmentation with directly trained SNNs is to empower the network with sufficient feature representation. Correspondingly, deep training structure is necessary. Previous attempts at deep direct training of SNNs mainly focus on classification tasks. For instance, Fang et al. Fang, Yu, Chen, Huang, et al. (2021) and Hu et al. Hu, Wu, Deng and Li (2021) proposed SEW-ResNet and MS-ResNet, respectively. They advanced SNNs to be trained directly on more than one hundred layers deep networks without the gradient vanishing/exploding problem. However, when applying their structures towards semantic segmentation, two major problems arise.

Firstly, there is a concern about achieving full spiking in the network to maximize energy efficiency, as hybrid models incorporating non-spiking operations, such as the multiply-accumulation (MAC) operations introduced in SEW-ResNet (Fig. 1.a), may undermine the low-power consumption property of SNNs. Additionally, the presence of non-spiking operations in SEW-Block poses challenges for the deployment of neuromorphic chips. Several neuromorphic chips only support spike operations, making it difficult to directly deploy hybrid models that incorporate non-spiking operations (Davies et al., 2018; Frenkel, Legat, & Bol, 2019). This limitation hinders the practical implementation of SEW-Block on such neuromorphic hardware platforms. Secondly, there is a computational overhead issue. SEW-ResNet and MS-ResNet (Fig. 1.b) both extract features by directly stacking 3×3 convolutional blocks, resulting in high memory overheads and computational complexity. It is crucial to explore methods that fully exploit the computational efficiency of SNNs to overcome these challenges.

To address the aforementioned challenges and enable directly trained SNNs to exhibit effectiveness and efficiency in semantic segmentation, this paper introduces the following key contributions:

Firstly, we propose a novel **M**ulti-scale and **F**ull-**S**pike Semantic **Seg**mentation network (MFS-Seg), which represents the first attempt to implement semantic segmentation based on deep direct training of SNNs. The MFS-Seg adopts a coarse-to-fine strategy for feature extraction, drawing inspiration from the multi-input multi-output (MIMO) UNet framework used in deblurring networks (Cho, Ji, Hong, Jung, & Ko, 2021). By incorporating multiple cascaded U-Nets within a single U-shaped network, the multi-scale U-Net mechanism enhances the representation capability of shallow networks, thereby improving the performance of the full spike pattern. Leveraging information flow at different scales, our MFS-Seg achieves high performance within a remarkably short time step of only 5.

Furthermore, towards deep training, we design an **E**fficient **F**ull-**S**pike **Res**idual Block (EFS-Res) that enables the model to better extract object features. We also demonstrate that EFS-Res could achieve deep training while avoiding the spike degradation problem based on the block dynamical isometry. Our specially designed EFS-Res utilizes depthwise separable convolution to improve the distributions of spiking feature maps, leading to more precised segmentation edges. While improving performance, these mechanisms also make the model lightweight in a full spike pattern which denotes significant energy-saving benefits. EFS-Res is embedded in the MFS-Seg network architecture, and by visualizing the spike distribution, we can find that the spikes at the edges of the object are more informative which shows the effectiveness of our model.

Finally, we validate on the frame-based camvid dataset that our model could achieve comparable performance compared to the mainstream ANN based UNet model. On event-based DDD17 datasets and DSEC-Semantic dataset, our approach achieves high performance with low power consumption. Our model reduces the memory overhead by **31** times and the energy consumption by **13** times.

In conclusion, our motivation of this paper is to provide an efficient and effective solution for semantic segmentation based on deep direct training of SNNs. We believe that our network holds potential for future deployment on neuromorphic chips, given its full spike nature and ability to avoid unnecessary computations.

The remainder of this article is organized as follows: In Section 2, we present a comprehensive overview of related work in the field of semantic segmentation. Section 3 introduces the neurons of SNN, training strategies, and energy consumption calculation methods, providing the necessary background for our proposed approach. In Section 4, we provide a detailed description of the EFS-Block and MFS-Seg structures, outlining the key components of our proposed network. Section 5 presents specific comparative experiments and visualizations to evaluate the performance and effectiveness of our approach. Finally, Section 6 concludes this work, summarizing our findings and discussing potential future directions for research and application.

2. Releated work

2.1. Effective spiking neural networks

Performance improvement tends to take deeper networks to enhance the representation. Towards deep training of SNNs mainly includes two strategies: ANN-to-SNN conversion and the directly trained SNNs. The main idea of ANN-to-SNN conversion is to approximate the average firing rate of SNNs to the continuous activation value of ANNs that use ReLU as the nonlinearity (Cao, Chen, & Khosla, 2015; Diehl et al., 2015). The trade-off between accuracy and latency has always constrained the development of ANN-to-SNN training strategies, since the elimination for approximation errors requires large time steps (Wu et al., 2021). Recent Spike Calibration (Li et al., 2022) can achieve comparable performance to ANN for semantic segmentation at a few hundred time steps, however, when the time step is less than 100, the performance decreases severely. In addition, bio-inspired event data, which can be effectively combined with neuromorphic hardware (Haessig, Cassidy, Alvarez, Benosman, & Orchard, 2018), are not applicable to the ANN-to-SNN conversion approach.

Directly trained SNNs are implemented using surrogate gradients (Neftci, Mostafa, & Zenke, 2019) for direct training. From the original SpikeProp (Bohte, Kok, & La Poutre, 2002) to the latest STBP (Wu, Deng, Li, Zhu, & Shi, 2018), the ongoing refinement of the gradient descent algorithms significantly improve the network accuracy. Diverse coding mechanisms such as time-coding (Comsa et al., 2020) and rate-coding (Fang, Yu, Chen, Masquelier, et al., 2021) enable directly trained SNNs to work well at short time steps. To overcome the problem of gradient vanishing or explosion, the proposed threshold-dependent batch normalization (TDBN) (Zheng et al., 2021) effectively expands the SNN from a shallow structure (<10 layers) to a deep structure (50 layers). Hu et al. Hu, Wu, et al. (2021) and Fang et al. Fang, Yu, Chen, Huang, et al. (2021) further modified the residual structure to advance the directly trained SNNs for over 100 layers on the classification task. Also deep SNNs have been tried on other tasks such as image/video reconstruction task (Ran et al., 2021; Zhu et al., 2022), multimodal pattern reconstruction (Xu et al., 2021) and object tracking (Zhang et al., 2022), etc. Recently, although there are some works based on directly trained SNNs explored on the dense-level image segmentation task (Kim et al., 2022; Kirkland, Di Caterina, Soraghan, & Matich, 2020), their structures are so shallow that they perform excessively poorly, reducing the effectiveness of direct training of SNNs.

2.2. Computational efficient segmentation

The rapid development of deep learning advances the semantic segmentation task towards high performance. While most of current models enhance performance at the expense of computational efficiency. In embedded systems or hardware deployments, high energy consumption and excessive memory overhead would limit the application of the models. Traditional artificial neural network-based

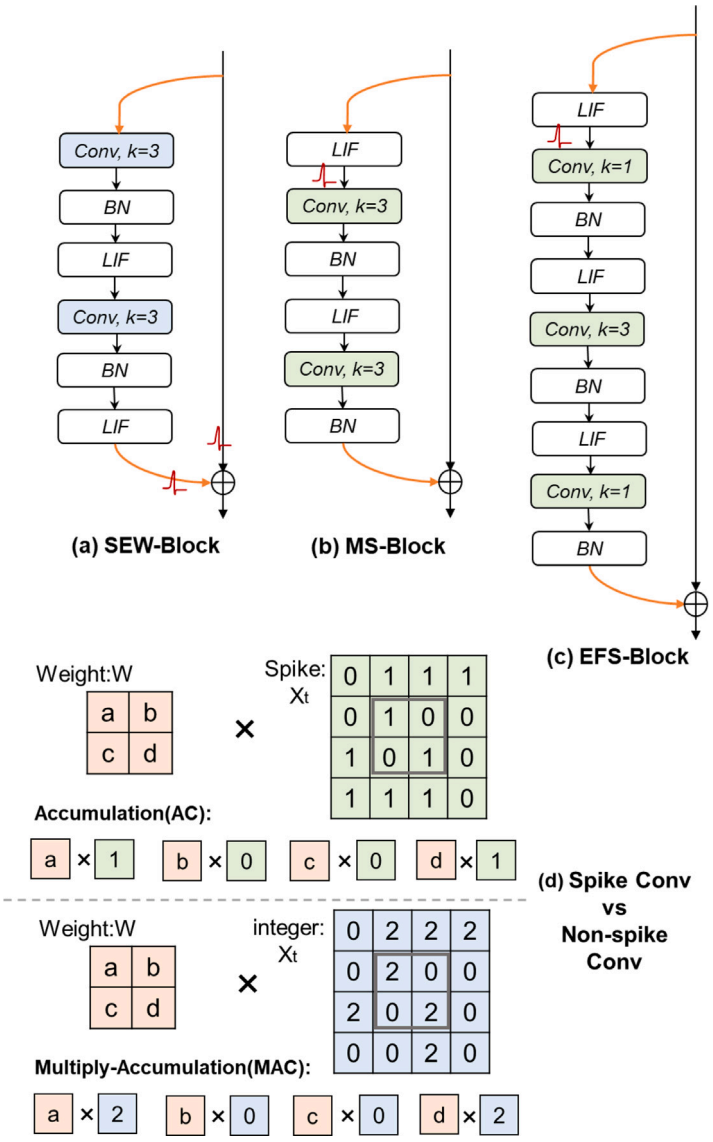


Fig. 1. Comparison of Mainstream Spike Residual Blocks. (a) The sum of spikes in SEW-ResNet causes non-spike convolution operations. (b) High computational complexity of MS-ResNet. (c) The structure of our proposed Efficient Full Spike Residual Block (EFS-Res). (d) Illustration of the Spike Convolution and Non-Spike Convolution operations.

solutions mainly include network pruning (Han, Mao, & Dally, 2015; Han, Pool, Tran, & Dally, 2015; He, Zhang, & Sun, 2017), knowledge distillation (Chen et al., 2019; Han et al., 2018; Hinton, Vinyals, & Dean, 2015), quantization (Yang, Deng, Yang, Xie, & Li, 2021), and lightweight module design. The first two optimize the model structure by post-processing approaches only after the whole model is completed training. The latter makes it possible to train deep neural networks directly on mobile terminals, and the typical modules include ShuffleNet (Ma, Zhang, Zheng, & Sun, 2018; Zhang, Zhou, Lin, & Sun, 2018), MobileNet (Howard et al., 2017; Sandler, Howard, Zhu, Zhmoginov, & Chen, 2018), and SqueezeNet (Iandola et al., 2016). All of these approaches work well to address the memory overhead problem, but not the high energy consumption problem. As ANNs transmit floating-point signals would introduce multiply-accumulation (MAC) operations, which result in more computational complexity and more energy cost.

Some explorations of SNN-based semantic segmentation have attempted to provide more energy-efficient solutions. The previous work attempted with ANN-to-SNN conversion (Baltes, Abujahar, Yue, Smith, & Liu, 2023; Cheni, Rueckauer, Li, Delbruck, & Liu, 2021; Li et al., 2022; Patel et al., 2021) which requires long inference time and cannot be applied to event camera data due to the inherent limitations of the approach. Attempts (Kim et al., 2022; Kirkland et al., 2020) based on direct training of SNNs on semantic segmentation are currently ineffective as the structures are quite shallow, while with high memory overhead. Considering that SNNs transmit spike signals, their inherent boundedness makes them well-suited for addressing the pixel-level task of image segmentation.

Designing SNN architectures that leverage these inherent characteristics is a significant research challenge. In the field of semantic segmentation, popular architectures include the Encoder-Decoder (Badrinarayanan, Kendall, & Cipolla, 2017; Ronneberger, Fischer, & Brox, 2015) and Encoder (Chen, Zhu, Papandreou, Schroff, & Adam, 2018; Sun et al., 2019) structures. Existing SNN structures for semantic segmentation tasks primarily rely on Encoder-based designs (Kim et al., 2022; Kirkland et al., 2020). Several network architectures, such as encoder structures (Qammaz & Argyros, 2019; Xu et al., 2023, 2022) and auto-encoders (Kamata, Mukuta, & Harada, 2022; Xu et al., 2021), have been proposed to enhance the feature extraction capabilities of SNNs for various tasks. In our work, we aim to design a comprehensive spike model and adapt the spike distribution to significantly improve the computational efficiency of directly trained SNNs for semantic segmentation.

2.3. Vision sensors for segmentation

Significant vision sensors in semantic segmentation include frame-based and event-based cameras (Wu et al., 2022). The majority of current works (Guo, Liu, Georgiou, & Lew, 2018) for semantic segmentation are proposed for common frame cameras which are sensing at the fixed frame rate and present some limitations in challenging scenarios (e.g., fast motion, over-exposure, and low light). Bio-inspired event cameras (e.g., DVS (Hu, Liu, & Delbruck, 2021), ATIS (Posch, Matolin, & Wohlgenannt, 2010), and DAVIS (Binas, Neil, Liu, & Delbruck, 2017)) have appeared and captured the interest of their advantages: high temporal resolution (microseconds), high dynamic range (up to 120 dB), low redundancy, and low power consumption. Spatio-temporal representation and exploiting rich temporal cues from asynchronous events are mainly based on the ANNs (Alonso & Murillo, 2019a; Jia et al., 2023; Sun, Messikommer, Gehrig, & Scaramuzza, 2022). Motivated by the temporal dynamics of the SNNs, the Spiking-DeepLab and Spiking-FCN (Kim et al., 2022) attempt to achieve comparable performance to ANN networks on shallow networks. However, the strengths of SNNs for asynchronous event data are not fully exploited due to the flaws of the shallow structure of their networks. Therefore, we propose a deeply trainable SNN that performs well on both frame-based and event-based data.

3. Preliminary

3.1. Preliminary of SNNs

Spiking Neuron. Human vision has its own ability to distinguish between different objects, even in the face of multiple foreground or background distractions. Spiking neural networks, as biologically plausible networks, mimic the human brain in transmitting a sequence of binary signals and converting them into valuable information. Compared with ANNs, which focus only on information in the spatial domain, SNNs also have temporal dynamics. Typically, the Leaky Integrate-and-Fire (LIF) model (Abbott, 1999), the Hodgkin–Huxley (H-H) model (Hodgkin & Huxley, 1952) and the Izhikevich model (Izhikevich, 2003) are the most prominent spiking neuron models. In particular, the LIF model for its optimal trade-off on computational complexity and biological plausibility is widely used (Li et al., 2023). In our work, we use the iterative LIF model proposed by Wu et al. Wu et al. (2019), which can be described as:

$$V_i^{t+1,n+1} = \tau V_i^{t,n+1}(1 - X_i^{t,n+1}) + \sum_j W_{ij}^n X_j^{t+1,n} \tag{1}$$

$$X_i^{t+1,n+1} = H(V_i^{t+1,n+1} - V_{th}) \tag{2}$$

where the $V_i^{t,n+1}$ represents the membrane potential of the ith neuron in the $n+1$ layer at the t timestep, τ is defined as a decay factor for leakage. The weighted W_{ij}^n sum of j spikes $X_j^{t+1,n}$ from the previous layer n is transmitted to the synaptic input of the current layer. $H(\cdot)$ represents the Heaviside step function which satisfies $H(x) = 1$ for $x \geq 0$, otherwise $H(x) = 0$. Neuronal firing spiking activity is regulated by thresholds V_{th}, and the $V^{t+1,n+1}$ will be reset to V_{rest} once the neuron emits a spike at the $t+1$ time step.

Training Strategy. As for the training strategy, considering the non-differentiability of spikes in backpropagation, we use the surrogate gradient backpropagation mechanism which can be represented as:

$$\frac{\partial X_i^{t,n}}{\partial V_i^{t,n}} = \frac{1}{a} sign(|V_i^{t,n} - V_{th}| \leq \frac{a}{2}) \tag{3}$$

where a acts as a regulatory factor to ensure the integral of the gradient is 1 and determines the curve steepness.

We consider both spatial and temporal domain and adopt the TDBN (Zheng et al., 2021) normalization method. The TDBN can be represented as:

$$V_i^{t+1,n+1} = \tau V_i^{t,n+1}(1 - X_i^{t,n+1}) + \mathrm{TDBN}(I_i^{t+1}) \tag{4}$$

$$\mathrm{TDBN}(I_i^{t+1}) = \lambda_i \frac{\alpha V_{th}(I_i^{t+1} - \mu_{ci})}{\sqrt{\sigma_{ci}^2 + \epsilon}} + \beta_i \tag{5}$$

where μ_{ci}, σ_{ci}^2 represent the mean and variation values for every channel using a mini-batch of sequential inputs $\{I_i^{t+1} = \Sigma_j W_{ij}^n X_j^{t+1,n} | t = 0, \ldots, T-1\}$, ϵ represents a tiny constant to avoid dividing by zero, λ_i, β_i are two trainable parameters, and α is a threshold-dependent hyper-parameter.

3.2. Energy consumption

In this work, we compare the energy consumption of ANNs and SNNs and assume that the data for various operations are 32-bit floating-point implementation in 45 nm technology (Horowitz, 2014), where $E_{MAC} = 4.6pJ$ and $E_{AC} = 0.9pJ$.

ANNs. The number of operations is often used to measure the computational energy consumption of neuromorphic hardware. For ANNs, floating-point operations (FLOPs) involve multiply-and-accumulate (MAC) operations that $FLOPs = 2 \cdot MACs$. $MACs$ represent the total number of multiply-add operations.

Considering that the FLOPs in the whole network are basically generated by the convolutional modules, here we analyze the convolutional modules specifically. For the convolutional modules, the total number of operations can be represented separately with (w) or without (w/o) bias as

$$FLOPs_{Convs} = \begin{cases} 2 \cdot C_{in} \cdot k^2 \cdot H \cdot W \cdot C_{out}, & w \\ (2 \cdot C_{in} \cdot k^2 - 1) \cdot H \cdot W \cdot C_{out}, & w/o \end{cases} \tag{6}$$

where the C_{in}, C_{out} are input and output channels respectively, k is the size of the convolution kernel, and H and W denote the output image size. The energy consumption of ANN can be described as $E_{ANN} = 4.6 \cdot MACs$.

SNNs. Each spiking neuron emits only one spike and involves in accumulate operations (AC) in SNNs, the computational complexity can be expressed as $FLOPs = ACs$. ACs denote the total number of accumulate operations.

However, many current SNNs introduce additional MAC operations due to their design flaws (Fang, Yu, Chen, Huang, et al., 2021). Thus, we quantify the energy consumption of vanilla SNNs as E_{SNN}:

$$E_{SNN} = T \cdot (fr \cdot 0.9 \cdot ACs + 4.6 \cdot MACs) \tag{7}$$

where T and fr represents the total time steps and the block firing rate. When there is no additional MAC operation in the SNNs, the energy consumption ratio can be represented as

$$\frac{E_{SNN}}{E_{ANN}} = T \cdot fr \cdot 0.9 \cdot \frac{E_{AC}}{E_{MAC}} \tag{8}$$

4. Methodology

In this section, we first present a brief overview of the input format and whole structure of the network. Then, we present the details of how the multi-scale encoder–decoder to extract segmented object features. Finally, we illustrate the Efficient Full Spike Residual Block which is the key component of the encoders and decoders, accompanied by a theoretical analysis and proof of its gradient stability.

4.1. Network input

Event-based streams. Given the spatio-temporal window Γ, the asynchronous event stream $E = \{e_n \in \Gamma : n = 1, \ldots, N\}$ represents a sparse grid of points in 3D space. In particular, an event $e_n = (x_n, y_n, t_n, p_n)$ is generated for a pixel (x_n, y_n) at the time step t_n when the logarithmic light change $I(x, y, t)$ exceeds the threshold θ_{th}. The

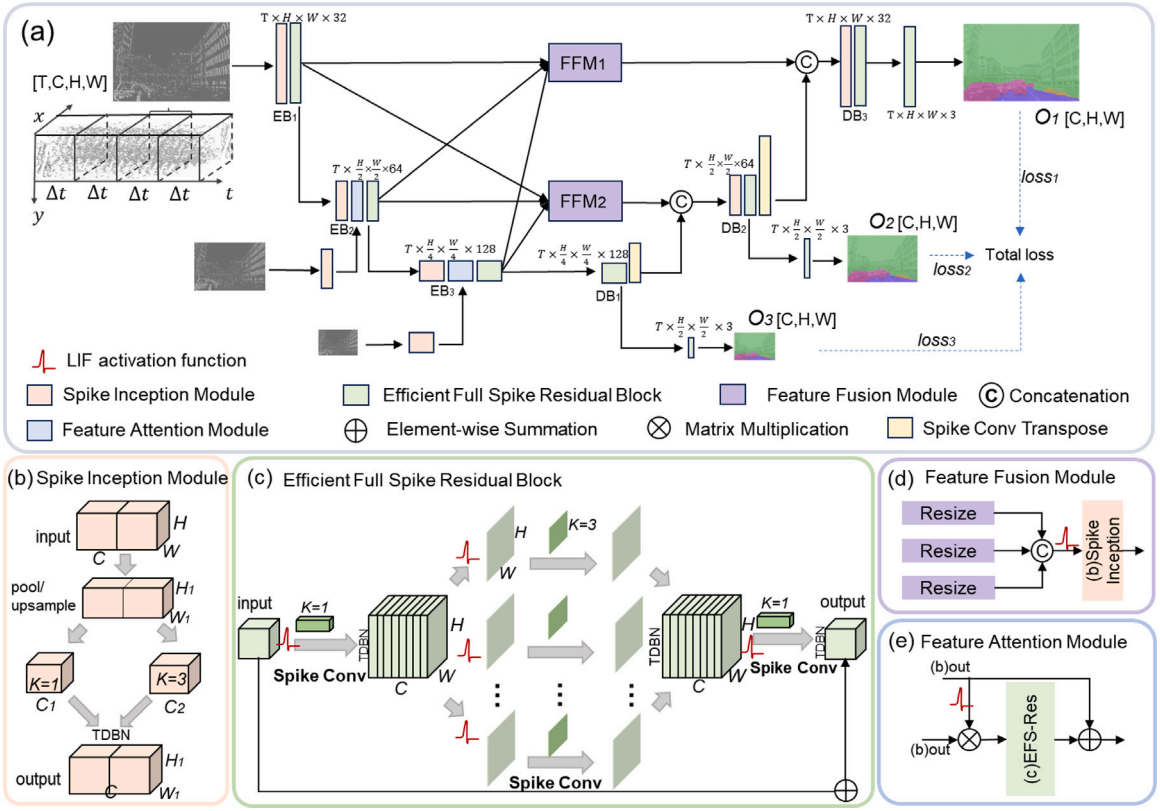


Fig. 2. Multi-scale Full Spike Semantic Segmentation Model (MFS-Seg). (a) Our design of a multi-scale full spike semantic segmentation model.(b)Spike Inception module to decouple feature representations. (c) Detailed illustration of the signal transmission of our proposed EFS-Res. (d) Feature fusion at different scales. (e)Attention mechanisms for different scale features.

polarity $p_n \in \{-1, 1\}$ denotes the increase or decrease of light intensity. In this work, we follow the handling of the DDD17 dataset in the ESS model (Sun et al., 2022) to encode each event sequence as several temporal bins which can be represented as histograms. Typically, the whole event stream E can be split into a number of small event sequences based on a constant temporal window dt. Our SNN model with spatio-temporal dynamics processes T fixed time steps each time, and the total input sequence can be represented according to the manner of dividing the event sequences as $\Gamma = T \times dt$.

Frame-based streams. Normally, considering the spatio-temporal feature of SNNs, the static images generated by the frame cameras are copied and utilized as the input frame for each time step (Yao, Hu, et al., 2023; Yao, Zhao, et al., 2023).

4.2. Network overview

In this work, our goal is to predict the class of each pixel from the input static images or event streams that can be represented as $\mathbf{X} = \{X_t\}_{t=1}^{T}$ to implement semantic classification of objects. We propose the deep direct training SNN architecture for semantic segmentation, namely MFS-Seg. As shown in Fig. 2, this network is essentially a variant UNet that mainly includes a multi-input encoder, a multi-output decoder. The multi-input encoder and multi-output decoder are composed of three encoder blocks (EBs) and three decoder blocks (DBs) respectively. The Spike Inception Module and our proposed Efficient Full Spike Residual Block in this multi-scale structure contribute remarkably to the extraction of segmentation features. The Feature Fusion Module (FFM) and the Feature Attention Module (FAM) are designed to fuse features at different scales.

The final output of the multi-scale images is directly compared to the corresponding resized original images for computing the cross-entropy loss.

4.3. Multi-scale and full spike model

The encoder block (EB) is mainly composed of three parts: the Spike Inception Module, Feature Attention Module and our proposed Efficient Full Spike Residual Block. The features of all the encoder blocks are fused through FFM and fed to the decoder blocks. The decoder block (DB) mainly contains the Spike Inception Module, Efficient Full Spike Residual Block and the Spike Transposed Convolution used for resizing feature dimensions.

Multi-Input Encoder. Firstly, all the input datas are encoded into membrane potential signals by the Spike Inception Module. As shown in Fig. 2.b, the Spike Inception Module replaces the traditional 3×3 convolution block with a 1×1 and a 3×3 convolution block. We employ the idea of inception (Szegedy, Vanhoucke, Ioffe, Shlens, & Wojna, 2016) to produce a representation of mutually decoupled features by activating more output branches, which could be attention to the details of the edges of the segmented objects.

Then, the features of the previous scale (EB_{n-1}) are emphasized or suppressed in the Feature Attention Module (see Fig. 2.e). In the Feature Attention Module, we use the maxpool to change the dimension and a convolution layer to change the number of channels of the last EB. The LIF activation function is used to activate the last layer of features, and the scale fusion is done by matrix multiplication, which is essentially a multiplication of spike and still will not introduce additional MAC operations.

Finally, after the weighted upward encoded features, the fused features are fed into our proposed Efficient Full Spike Residual Block (see Fig. 2.c). The number of blocks can be dynamically modified depending on different tasks. For the features extracted from different encoder blocks, we directly concatenate and scale the output to the matching decoder block using the Feature Fusion Modules (shown in Fig. 2.d).

Multi-Output Decoder. Given the ability of these multi-scale feature maps to directly mimic image patterns, we generate segmented frames of multiple sizes O that correspond to different inputs which can be described as:

$$O = \begin{cases} D(DB_n(EB_n^{out}); FFM_n^{out}), & n = 1, 2 \\ D(DB_n(EB_n^{out})), & n = 3 \end{cases} \tag{9}$$

where the $FFM_n^{out}, EB_n^{out}, DB_n^{out}$ are respectively the outputs of the nth level FFM, EB, DB. We revert the feature map to the same size as the input frames using the Spike Transposed Convolution for multi-scale feature fusion.

The output of the DB is the multi-time step feature map rather than the final in-class frames. Mapping function D is required for generating an intermediate output image, where we use a spike-convolutional layer to transmit the membrane potential signals, and then output the final segmentation results by rate-coding (Hu, Wu, et al., 2021).

4.4. Efficient full spike residual block

Currently, the main structures used for deep training SNNs involve SEW-ResNet and MS-ResNet (Fig. 1). However, the SEW-ResNet essentially transmits a mixture of spikes and integers in the network, when both the residual path and shortcut path transmit spike signals, the addition operation would result in a non-spiking convolution operation in the next module. MS-ResNet ensures that the entire network is non-spiking by placing the LIF before the convolution. As shown in Fig. 1.b, the residual path consists of two 3×3 convolutions and we would analyze the computational overhead it yields, which is unsympathetic to the deployment of neuromorphic chips (Pei et al., 2019; Zhang et al., 2020).

FLOPs of MS-Block. For the calculation of convolutional FLOPs with bias, we define k as the size of the convolutional kernel, c_1, c_2 as the number of input channels and output channels respectively, typically these two are equal. $h \cdot w$ as the size of a new feature map generated by a convolution layer. The FLOPs incurred by MS-Block can be expressed as:

$$k^2 \cdot h \cdot w \cdot c_1 \cdot c_2 + k^2 \cdot h \cdot w \cdot c_2 \cdot c_2$$

where the k is 3. After simplification, we get $18 \cdot h \cdot w \cdot c_1^2$.

Depthwise Separable Convolutions are a key building block for many efficient neural network architectures in ANNs (Howard et al., 2017). Motivated by this, we propose an efficient fully spiked residual block (EFS-Res) which is applicable to SNNs as shown in Fig. 1.c. The residual path is mainly composed of three convolution blocks, where the first 1×1 convolution reduces the number of channels, the second depth-wise convolution applies group convolution to reduce the number of parameters, and the last 1×1 Pointwise convolution combine them to create new features. The LIF activation function is placed before each convolution to ensure that the whole block is spiking, and the TDBN is used to normalize the time-domain and spatial-domain information after convolution.

FLOPs of our EFS-Block. When we set the input channel and out channel of depth-wise convolution to $2 \cdot c_2$, the FLOPs for our EFS-Block can be represented as:

$$c_1 \cdot 2 \cdot c_2 \cdot k_1^2 \cdot h \cdot w + 2 \cdot c_2 \cdot 2 \cdot c_2 \cdot \frac{1}{g} \cdot k_2^2 \cdot h \cdot w + 2 \cdot c_2 \cdot c_2 \cdot k_1^2 \cdot h \cdot w$$

where the convolution kernel size k_1, k_2 are 1 and 3 respectively, the g denotes the group size of the group convolution that equals to $2 \cdot c_2$. After substitution, we get the FLOPs as $4 \cdot c_1^2 \cdot h \cdot w + 2 \cdot c_2 \cdot h \cdot w$.

Comparison. As can be seen, our EFS-Block reduces the computational overhead by about 4 to 5 times compared to MS-Block. Following experiments (Section 5) show that our EFS-Block is comparable to MS-Block on performance.

4.5. Analysis of gradient vanishing/explosion problems

To sufficiently illustrate that our EFS-ResNet can be trained deeply, we analyze and demonstrate theoretically that it could avoid the spike degradation problem in this section. Block Dynamical Isometry (Chen, Deng, Wang, Li, & Xie, 2020), which has been developed in recent years as a theoretical explanation of well-behaved neural network, measures the change of gradient norm in individual block.

Without loss of generality, a neural network can be viewed as a serial of blocks:

$$f(x_0) = f_{\theta^L}^L * f_{\theta^{L-1}}^{L-1} * \ldots * f_{\theta^1}^1(x_0), \tag{10}$$

Where θ^i is the parameter matrix of the ith layer. For simplicity, we denote $\frac{\partial f^i}{\partial f^{i-1}}$ as J_i, which means the Jacobian matrix of the block j, j is the index of the corresponding block.

Definition 1 (*Block Dynamical Isometry*)**.** Consider a neural network that can be represented as a series of blocks and the jth block's Jacobian matrix is denoted as J_j. If $\forall j; \phi(J_j J_j^{\mathrm{T}}) \approx 1$ and $\varphi(J_j J_j^{\mathrm{T}}) \approx 0$, the network achieves "Block Dynamical Isometry" and can avoid gradient vanishing or explosion.

Here, ϕ means the expectation of the normalized trace, φ means $\phi(\boldsymbol{A}^2) - \phi^2(\boldsymbol{A})$. The theory ensures the gradient of the network will not decrease to 0 or explode to ∞ since every block have $\phi(J_j J_j^{\mathrm{T}}) \approx 1$. And $\varphi(J_j J_j^{\mathrm{T}}) \approx 0$ makes sure that the accident situation will not happen. And in most cases (Hu, Wu, et al., 2021; Zheng et al., 2021), $\phi(J_j J_j^{\mathrm{T}}) \approx 1$ is enough for avoiding gradient vanish or exploding. More detailed description of the notation and the theory are in Chen et al. (2020).

Lemma 1 (*Multiplication*)**.** *(Theorem 4.1 in Chen et al. (2020)) Given* $\boldsymbol{J} := \prod_{j=L}^{1} \boldsymbol{J}_j$, *where* $\{\boldsymbol{J}_j \in \mathbb{R}^{m_j \times m_{j-1}}\}$ *is a series of independent random matrices. If* $(\prod_{j=L}^{1} \boldsymbol{J}_j)(\prod_{j=L}^{1} \boldsymbol{J}_j)^T$ *is at least the 1st moment unitarily invariant, we have*

$$\phi\left((\prod_{j=L}^{1} \boldsymbol{J}_j)(\prod_{j=L}^{1} \boldsymbol{J}_j)^T\right) = \prod_{j=L}^{1} \phi(\boldsymbol{J}_j \boldsymbol{J}_j{}^T). \tag{11}$$

Lemma 2 (*Addition*)**.** *(Theorem 4.2 in Chen et al. (2020)) Given* $\boldsymbol{J} := \prod_{j=L}^{1} \boldsymbol{J}_j$, *where* $\{\boldsymbol{J}_j \in \mathbb{R}^{m_j \times m_{j-1}}\}$ *is a series of independent random matrices. If at most one matrix in* $\boldsymbol{J}_j$ *is not a central matrix, we have*

$$\phi(\boldsymbol{J}\boldsymbol{J}^T) = \sum_j \phi(\boldsymbol{J}_j \boldsymbol{J}_j{}^T). \tag{12}$$

Lemma 3. *For each of L sequential blocks in a neural network, we have* $\phi\left(\boldsymbol{J}_i \boldsymbol{J}_i^T\right) = \omega + \tau\phi\left(\tilde{\boldsymbol{J}}_i \tilde{\boldsymbol{J}}_i{}^T\right)$ *where* $\boldsymbol{J}_i$ *is its Jacobian matrix,* ω *and* τ *are variables. Given* $\lambda \in \mathbb{N}^+ < L$, *if* $C_L^\lambda(1-\omega)^\lambda$ *and* $C_L^\lambda \tau^\lambda$ *are small enough, the network would be as stable as a* λ*-layer network when the network satisfies* $\forall i, \phi\left(\boldsymbol{J}_i \boldsymbol{J}_i^T\right) \approx 1$.

Proposition 1. *The EFS-Block can be stable as a* λ*-layer network which satisfies* $\phi(\boldsymbol{J}_j \boldsymbol{J}_j^T) = 1$ *and* $\lambda \in \mathbb{N}^+ < L$.

Proof. For the EFS-Block, we denote the Jacobian matrix of the residual path as $\tilde{\boldsymbol{J}}_i$. According to Lemma 3, we have $\phi\left(\boldsymbol{J}_i \boldsymbol{J}_i^T\right) = 1 + \tau\phi\left(\tilde{\boldsymbol{J}}_i \tilde{\boldsymbol{J}}_i{}^T\right)$, where γ is from the linear transformation $\gamma x + \beta$ with the normalization at the end of the residual path. EFS-ResNet can be viewed as an extreme example of Lemma 1 with $(1-\omega) \to 0$. Therefore, $\forall \lambda, C_L^\lambda(1-\omega)^\lambda$ is close to zero, and $C_L^\lambda \gamma^\lambda$ can be small enough for a given λ if γ is initialized as a relative small value. In this way, the error of non-optimal block will be influenced only within λ layers and the EFS-ResNet will be as stale as a much shallower λ-layer network.

5. Experiments

In this section, we evaluate the effectiveness and efficiency of our model on different sensor-generated datasets. We conduct experiments on the frame-based Camvid dataset (Brostow, Fauqueur, & Cipolla, 2009), and the event-based DDD17 dataset (Binas et al., 2017), the DESC-Semantic dataset (Sun et al., 2022).

Metrics. We utilize the standard metric *Mean Intersection over Union (mIoU)*, which is commonly used on the image segmentation task, for measuring the effectiveness of the model. The mIoU is calculated per class as:

$$mIoU(y,\bar{y}) = \frac{1}{C}\sum_{j=1}^{C}\frac{\sum_{i=1}^{N}\delta(y_{i,c},1)\delta(y_{i,c},\bar{y}_{i,c})}{\sum_{i=1}^{N}max(1,\delta(y_{i,c},1)+\delta(\bar{y}_{i,c},1))} \tag{13}$$

where δ is the Kronecker delta function, y_i denotes pixel i that belongs to the same class y, and $y_{i,c}$ represents the boolean that whether the pixel i is in a certain class c.

For the evaluation of model efficiency, we measure the number of parameters of to evaluate the memory usage (*Params*) and the number of float-point operations ($FLOPs$) to evaluate the computational complexity as defined in Section 3.2. Furthermore, we also measure the energy consumption of the model according to the description of Section 3.2.

Datasets. The Cambridge-driving Labeled Video Database (CamVid) contains over 700 images specified manually and provides ground truth labels that associate each pixel with one of 32 semantic classes. The dataset is spilt as 367 training pairs, 101 validation pairs and 233 test pairs following the general setting (Brostow, Shotton, Fauqueur, & Cipolla, 2008) and we annotate the ground truth to 11 semantic categories due to the rare occurrence of the remaining classes following the general setting. Considering the less data available in the training set, we merged the training and test sets and evaluated them on the validation set.

The DDD17 dataset, as the first ever public dataset of real automotive end-to-end training data, is recorded on different scenarios (e.g., motorways and urban scenarios) with an advanced 346 × 260 pixel DAVIS sensor (Brandli, Berner, Yang, Liu, & Delbruck, 2014). It provides synchronized grayscale and event-based information, while it does not provide semantic segmentation labels. Therefore, we use the pseudo-labels offered by the Ev-SegNet model (Alonso & Murillo, 2019b) which are generated by a pre-trained network running on the grayscale frames of the DAVIS346B (Brandli et al., 2014). Due to the low resolution of the DAVIS346B, multiple classes are merged and the granularity of the labels are reduced.

The DSEC-Semantic dataset (Sun et al., 2022) is an extension of the sequences of the large-scale DSEC dataset (Gehrig, Aarents, Gehrig, & Scaramuzza, 2021), consisting of recordings captured in both urban and rural environments. This dataset provides 640 × 440 pixel labels for 11 different classes, namely background, building, fence, person, pole, road, sidewalk, vegetation, car, wall, and traffic sign. Notably, the labels in the DSEC-Semantic dataset exhibit superior quality and offer more intricate details compared to the labels found in DDD17.

Implementation Details. We perform the experiments with the EFS-Block explained in Section 4.4 and the MFS-Seg framework detailed in Section 4.3. We set the reset value V_{reset} of LIF neurons to 0, the membrane time constant τ to 0.25, the threshold V_{th} to 0.5, and the coefficient α to 1. We train all model variations from scratch using the Adam optimizer. The network is trained for 600 epochs which are sufficient for convergence on one NVIDIA RTX3090 GPU for the Camvid dataset with a batch size of 2. With an initial learning rate of 1e−3, it is decreased by the factor of 0.1 at every 200 epochs. On the DDD17 dataset, we train the model for 100 epochs, with the batch size of 8 on one NVIDIA A100 GPU. The learning rate is initially set to 1e−3 and decreases by the factor of 0.1 at every 20 epochs. For the DSEC-Semantic dataset, we conduct experiments on the NVIDIA A100 GPU. The initial learning rate is set to 2e−3, the batch size is 4, and we train for 100 epochs. At the 60th epoch, we apply a learning rate decrease with a decay factor of 0.5.

The initial size of the static image is 260 × 346 and we transform it to 180 × 240. The original size of the DDD17 dataset images is transformed to 260 × 352. After applying event transformation, the DSEC-Semantic dataset is resized to images with dimensions of 440 × 640. We maintain the same settings as described in this article (Sun et al., 2022) to ensure consistency. The input sizes of the three encoder blocks are the original, twice downsampled and 4 times downsampled. For the evaluation of the model computational complexity, we calculate the total number of FLOPs of these networks using torchstat.[1]

5.1. Static image segmentation

Here we experimentally set the number of EFS-Blocks in each encode and decode block to 4. The whole network can reach a depth of at least 80 layers.

Effectiveness. The results demonstrate that the directly training method can achieve higher performance at only 5 time steps (presented in Table 1). Compared to the typical ANN methods such as UNet (Ronneberger et al., 2015), SegNet (Badrinarayanan et al., 2017) method based on encoder–decoder structure, and FCN (Long et al., 2015) with encoder structure, our model achieves better performance with the mIOU of **0.621**. When the UNet structure is implemented with SNN, the performance drops a lot, whereas our approach is comparable to the performance of the same structure ANN. When the model structures are all based on MFS-Seg framework, our efficient full-spike Block (EFS-Block) can also achieve comparable performance compared to the MS-Block, which illustrates the potential of the full spiking network to be further exploited.

Efficiency Analysis. From Table 1, it is well observed that at comparable performance, our EFS-Block reduce both the number of parameters and FlOPs by at least **3**× when comparing to the previous MS-Block. The firing rate of our model is slightly increased, which may be an effect of the more dense representation of features needed with fewer number of parameters. Nevertheless, it does not limit the energy advantage of our model and still reduces energy consumption by 2 times. The MFS-Seg structure better improves the model performance while bringing some computational complexity by certain inception module transformations. Fortunately, our EFS-Block reduces the computational complexity and requires only **1.11**M parameters and **4.43** GFLOPs. More importantly, our solution not only reduces the model complexity, but also enables an efficient **13**× reduction in energy consumption compared to the traditional ANN-based UNet structure.

5.2. Event-based segmentation

Our model adopts the data processing strategy of the ESS model (Sun et al., 2022). The dt of the raw event stream can be split into 50 ms, 250 ms. Alternatively, we can fix the event time intervals based on the number of events. The result we give utilizes networks that are more than 80 layers deep and the number of EFS-Blocks in each encode and decode block is set to 3 in DDD17 dataset and 4 in DSEC-Semantic dataset.

Effectiveness. Currently, methods based on directly training SNNs are in the early exploration stage for semantic segmentation tasks. Previous SNN-based approaches mainly include Spiking-DeepLab, Spiking-FCN (Kim et al., 2022) and SCGNet (Zhang et al., 2023). The architecture of Spiking-DeepLab and Spiking-FCN, are extremely shallow, which is not sufficiently to invoke the effectiveness of the directly trained SNNs. Overlaying network layers directly on these direct-connected structures would not improve performance and even yield

[1] https://github.com/Swall0w/torchstat

Table 1
Results on the frame-based Camvid dataset and the event-based DDD17 dataset and DSEC-Semantic dataset.

Dataset	Method	Model	Params(M)	GFLOPs	FR[a]	T	mIoU	Etotal(mJ)
Camvid	ANN	UNet (Ronneberger et al., 2015)	31.04	30.22	–	–	0.610	130.01
		FCN (Long, Shelhamer, & Darrell, 2015)	13.50	3.04	–	–	0.482	13.98
		SegNet (Badrinarayanan et al., 2017)	1.43	107.50	–	–	0.601	494.50
		MIMO UNet (Cho et al., 2021)	3.79	22.17	–	–	0.591	101.98
		MFS-MS[b] (Hu, Wu, et al., 2021)	3.75	24.45	–	–	0.632	112.47
		MFS-EFS[c]	1.11	8.85	–	–	0.633	40.73
	SNN	UNet (Ronneberger et al., 2015)	31.03	30.18	0.244	5	0.430	33.14
		spiking-FCN (Kim et al., 2022)	13.57	9.53	–	5	0.425	–
		MFS-MS	3.75	12.23	0.179	5	0.627	21.88
		MFS-EFS (ours)	1.11	4.43	0.243	5	**0.621**	**10.00**
DDD17	ANN	EV-SegNet (Alonso & Murillo, 2019a)	29.09	73.62	–	–	0.548	338.65
		E2ViD (Rebecq, Ranftl, Koltun, & Scaramuzza, 2019)	10.71	16.65	–	–	0.448	76.59
		ViD2E (Gehrig, Gehrig, Hidalgo-Carrió, & Scaramuzza, 2020)	29.09	73.62	–	–	0.560	338.65
		DTL (Wang, Chae, & Yoon, 2021)	60.48	16.74	–	–	0.588	77.01
		EvDistill (Wang, Chae, Yoon, Kim & Yoon, 2021)	59.34	12.45	–	–	0.580	57.27
		EV-Transfer (Messikommer, Gehrig, Gehrig, & Scaramuzza, 2022)	7.37	7.88	–	–	0.149	36.25
		ESS (Sun et al., 2022)	12.91	14.22	–	–	0.614	65.41
		EvSegFormer (Jia et al., 2023)	24.20	31.20	–	–	0.526	143.52
	hybrid[d] (ANN+SNN)	HALSIE (Biswas, Kosta, Liyanagedera, Apolinario, & Roy, 2022)	1.82	4.11	–	–	0.606	17.89
	SNN	spiking-DeepLab (Kim et al., 2022)	4.14	54.34	–	20	0.337	48.91
		spiking-FCN (Kim et al., 2022)	13.57	3.04	–	20	0.342	–
		SCGNet-S (Zhang, Fan, & Zhang, 2023)	0.49	–	–	4	0.493	–
		SCGNet-L (Zhang et al., 2023)	1.85	–	–	4	0.514	–
		MFS-MS	2.97	20.82	0.187	2	0.632	17.57
		MFS-EFS (ours)	0.92	8.11	0.238	2	**0.628**	**7.72**
DSEC-Semantic	ANN	EV-SegNet (Alonso & Murillo, 2019a)	29.09	405.18	–	–	0.518	1863.83
		E2ViD (Rebecq et al., 2019)	10.71	90.65	–	–	0.407	416.99
		EV-Transfer (Messikommer et al., 2022)	7.37	42.93	–	–	0.232	197.48
		ESS (Sun et al., 2022)	12.91	77.46	–	–	0.454	356.32
	SNN	spiking-FCN (Kim et al., 2022)	13.57	104.81	0.204	2	0.437	37.73
		MFS-EFS (ours)	1.11	28.88	0.245	2	0.461	**12.74**

[a] FR represents the firing rate.
[b] MFS- denotes the MFS-Seg framework. MS represents the spike residual module is MS-Block.
[c] EFS represents our proposed EFS-Block.
[d] Hybrid indicates that this model structure utilizes a mix of ANNs and SNNs.

Table 2
Comparing the performance of different spike lightweight blocks on the Camvid dataset. All experiments are performed in the MFS-Seg architecture with 5 time steps.

Block	Params(M)	GFLOPs	Firing rate	mIOU
EFS-Res18	0.38	2.65	0.216	0.565
Spike Shuffle	0.39	2.21	0.311	0.485
EFS-Res72	1.11	4.43	0.243	0.621
Spike Fire	1.25	4.75	0.249	0.582

spike degradation problems (Zheng et al., 2021). As for the SNN-based state-of-the-art method, SCGNet, it is based on FCN structure. We achieved significantly better results than the comparison model on the DDD17 dataset, particularly when using shorter time steps (2 timesteps), all while utilizing fewer parameters. The main reason for our better experimental results can be attributed to the unique structure of our model, which is a variant of UNet. Our model incorporates multi-scale inputs and outputs, along with a jump-junction structure that effectively preserves spatial information.

As a new solution to implement deep SNN semantic segmentation, we achieve good performance **(mIoU=0.633)** on the DDD17 dataset and **(mIoU=0.461)** on the DSEC-Semantic dataset. Our model on the DDD17 dataset performs better than the current methods whether ANN or SNN based which well encourages the exploration of SNNs based work on event camera datasets. For a fair comparison, we chose the results where the labels are all frames for comparison on the DSEC-Semantic dataset. Our model could achieve comparable performance to the best current ANN model (Sun et al., 2022) with less parameters and only 2 time steps.

Efficiency Analysis. Our MFS-Seg model requires only **1.11**, which is the minimum memory occupation of all methods. Although the FLOPs of HALSIE are minor, the model has higher energy consumption since the model mostly involves ANN for feature extraction. Our model consumes only **7.72 mJ** of energy with 2 time steps on the DDD17 dataset, which is around **43.87** times less energy consumption than the most typical ANN-based EVsegNet model. Compared to other SNN methods, we are able to reduce the energy consumption by at least 6 times. Through experimental analysis and energy consumption comparison, we verify the efficiency of our full spike based semantic segmentation scheme.

5.3. Ablation study

Efficient Block Comparison. For ANNs, typical lightweight models include SqueezeNet, MobileNet and ShuffleNet, all of which have achieved great success. The SqueezeNet is mainly based on the proposed fire module, which consists of squeeze layer and expand layer. Squeeze layer performs 1×1 convolution, and expand layer concats the feature maps derived from 1×1 and 3×3 convolution. This model is like VGG's idea of stacked convolution, which can be difficult to train in depth. MobileNet adopts the depth-wise convolution to reduce the network weight parameters in place of the traditional convolution. Furthermore, point-wise convolution is used to obtain all the feature map information of the input layer. The ShuffleNet similarly applies depth-wise convolution, the difference is that it uses channel shuffle to form new feature maps to solve the problem of information non-flow caused by group convolution.

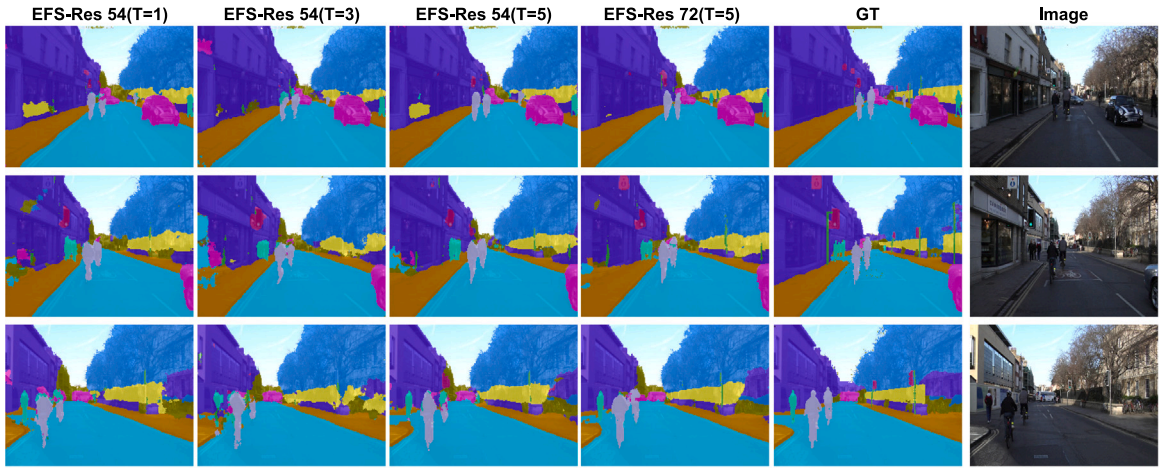


Fig. 3. Semantic Segmentation results on the Camvid Dataset. The first three columns are experimented on the same EFS-Res54 to illustrate that the longer the time step the better the performance. The comparison between the third and fourth columns shows that the deeper the depth of EFS-Res, the better the feature extraction capability. Res54 denotes that all EFS-Blocks in MFS-Seg contain 54 layers of convolution.

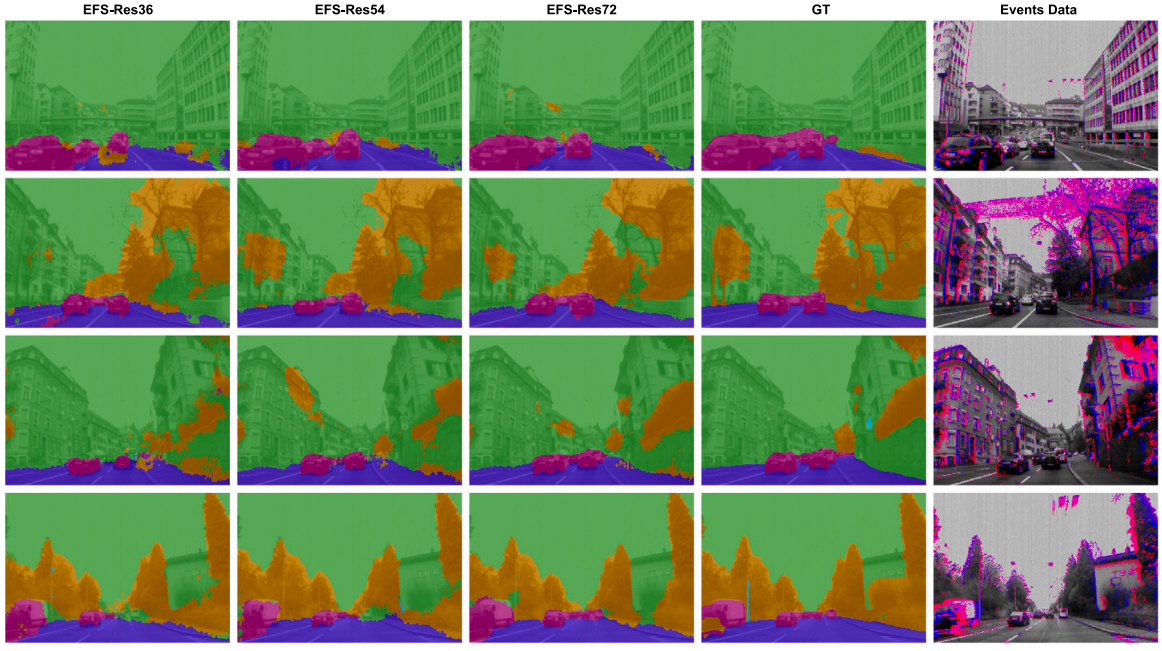


Fig. 4. Semantic segmentation results on the DDD17 dataset. The first three columns explore the effect of network depth on segmentation accuracy at the same time step. To visualize the event stream more clearly, we plot on the corresponding gray image.

Table 3
Efficiency analysis of different modules on Camvid dataset. w/o represents ablation experiments without corresponding modules. Incep denotes spike inception module. The group indicates that the EFS-Block is without the group convolutions for ablation experiments.

Module	Params (M)	GFLOPs	Firing rate	mIOU
MFS-Seg	0.92	3.83	0.240	0.611
w/o multi scale	0.92	3.83	0.240	0.602
w/o Incep	0.73	2.69	0.236	0.562
w/o group	6.28	19.61	0.180	0.623

Table 4
Ablation studies of different numbers of residual blocks on the Camvid dataset and the DDD17 dataset.

Dataset	Model	Params (M)	GFLOPs	FR	mIOU
Camvid	EFS-Res36	0.59	3.23	0.239	0.597
	EFS-Res54	0.92	3.83	0.240	0.611
	EFS-Res72	1.11	4.43	0.243	0.621
DDD17	EFS-Res36	0.74	6.83	0.234	0.609
	EFS-Res54	0.92	8.11	0.238	0.628
	EFS-Res72	1.11	9.38	0.227	0.640

We transfer the lightweight modules that are commonly used in ANN into SNN. To achieve deep training and full spiked, all of them have been redesigned shown in Fig. 6. All blocks are experimented with the MFS-Seg framework and the results are shown in the Table 2 on the Camvid dataset. Our EFS-Block is based on the structure of mobilenet V2, without the linear transformation, to ensure that the whole module is fully spiked. The performance of the other blocks are poor. Through the visualization of spiking attention distribution maps(shown in Fig. 5), we compare spike shuffleNet and EFS-Res18 with about the similar number of parameters, and it can be seen that the former object boundary feature extraction is very fuzzy. Moreover, with

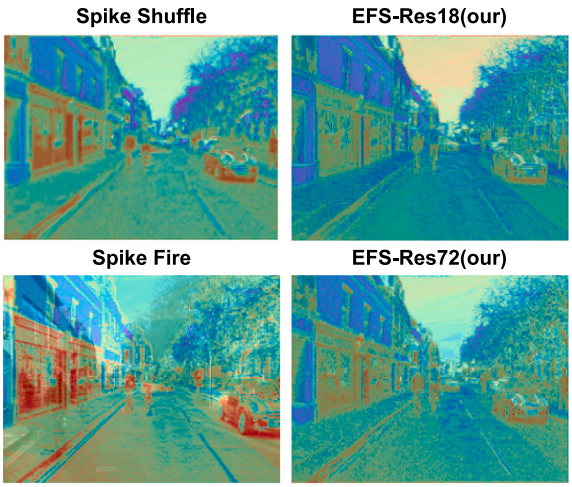


Fig. 5. Comparison of spike distribution maps for the spike lightening module. The corresponding rows of the models are compared in about the same number of parameters. The sharper the red edge represents the stronger the validity of the spike distribution and the better the feature extraction capability. The deeper red areas represent more intense spike activity, while purple areas indicate essentially no spike release activity.

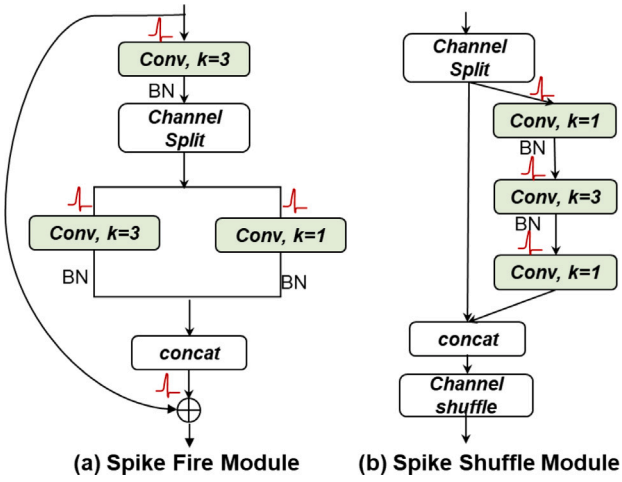


Fig. 6. Spike Fire Module and Spike Shuffle Module. We redesign the commonly used Fire and Shuffle modules in ANN by fully spiking them. BN represents the TDBN.

deeper layers of EFS-Res, the representation of object edge features will be more effective.

Module Efficiency Analysis. The differences of our MFS-Seg compared to previous architectures are mainly in the introduction of multi-scaling of inputs and outputs and Spike Inception modules. As well as the idea of group convolution is used in the design of EFS-Block to reduce the number of parameters. Here we analyze the effectiveness of each of these modules on the Camvid dataset. Each encode and decode module is set to 3, and the depth of the entire network is about 60 layers. We validate the effect of multi-scale feature extraction on model performance and simply set the error weights of the small and medium scales to 0. As seen in Table 3, the performance of the model is significantly reduced without the features of these two scales. For the Inception module, although it would cause an increase in FLOPs by a certain amount. However, there is a 50% increase in performance for this part of the computational complexity. For group convolution in EFS-Block, we verified its effectiveness by reducing 4× FLOPs and 3× number of parameters with comparable performance.

Table 5
Impact of the time step size on Camvid dataset.

T	1	3	5	7
Firing Rate	0.328	0.362	0.367	0.383
mIOU	0.554	0.580	0.611	0.618

Numbers of Residual Blocks. In Section 4.5, we theoretically analyze that our EFS-ResNet can achieve deep training. The depth of a single EFS-block is 3. Totally, MFS-Seg has 6 encode and decode modules. Here we set the number of each module to 2, 3, 4 respectively and report results in Table 4 on the Camvid dataset and DDD17 dataset based on EMS-Res36, EFS-Res54, and EMS-Res72. When the scale of the network is larger, the feature extraction ability becomes stronger (see Fig. 4).

Size of Time Steps. As the sparsity of event streams, different event sampling strategies would affect the ablation experiment of time steps on DDD17 dataset. Thus, we report the performance based on the Camvid dataset for $T = 1, 3, 5, 7$ in Table 5. Here we set each encode and decode module to work with 3, and the depth of the entire network to be about 60 layers. We show the segmentation results compared in Fig. 3, where it can be found that the accuracy of semantic segmentation is higher when the time step is longer. The time steps can be dynamically adjusted to achieve a balance of effectiveness and efficiency according to the needs of the actual task.

6. Conclusion

In this work, we have made pioneering contributions to the advancement of deep direct training-based SNNs for semantic segmentation. Inspired by the successful coarse-to-fine strategy used in deblurring tasks, we propose a novel multi-scale semantic segmentation network based on deep direct training of SNNs, achieving remarkable performance within an extremely short time step. To fully exploit the computational efficiency potential of SNNs, we introduce an efficient and full-spike module called EFS-Block. We demonstrate the module's capacity for deep training based on block dynamical isometry theory. Through comprehensive validation on the frame-based Camvid dataset, the event-based DDD17 dataset and the DSEC-Semantic dataset, we demonstrate the superiority of our model in terms of both effectiveness and efficiency compared to current models. Visualizing the spike distribution map reveals that spike activity is highly focused on edge features, further highlighting the effectiveness of our approach. Our network design leverages fully additive operations with minimal memory and computational overhead, effectively sparsifying and efficiently utilizing spiking activities. We firmly believe that our sparse fully spike network enables more efficient deployment of SNNs on neuromorphic chips.

CRediT authorship contribution statement

Qiaoyi Su: Investigation, Methodology, Project administration, Resources, Validation, Visualization, Writing – original draft, Writing – review & editing. **Weihua He:** Data curation, Investigation. **Xiaobao Wei:** Data curation, Formal analysis, Project administration. **Bo Xu:** Conceptualization, Project administration, Resources. **Guoqi Li:** Resources, Conceptualization, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Our code is available in https://github.com/BICLab/MFS-Seg.

Acknowledgments

This work was partially supported by National Science Foundation for National Science and Technology Major Project (2020AAA0105802), Distinguished Young Scholars (62325603), and National Natural Science Foundation of China (62236009, U22A20103, 62441606), and Beijing Natural Science Foundation for Distinguished Young Scholars (JQ21015).

References

Abbott, L. F. (1999). Lapicque's introduction of the integrate-and-fire model neuron (1907). *Brain Research Bulletin*, *50*(5–6), 303–304.

Alonso, I., & Murillo, A. C. (2019a). EV-SegNet: Semantic segmentation for event-based cameras. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*.

Alonso, I., & Murillo, A. C. (2019b). EV-SegNet: Semantic segmentation for event-based cameras. In *IEEE international conference on computer vision and pattern recognition workshops*.

Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(12), 2481–2495.

Baltes, M., Abujahar, N., Yue, Y., Smith, C. D., & Liu, J. (2023). Joint ANN-snn co-training for object localization and image segmentation. arXiv preprint arXiv:2303.12738.

Binas, J., Neil, D., Liu, S.-C., & Delbruck, T. (2017). DDD17: End-to-end DAVIS driving dataset. arXiv preprint arXiv:1711.01458.

Biswas, S. D., Kosta, A., Liyanagedera, C., Apolinario, M., & Roy, K. (2022). HALSIE–hybrid approach to learning segmentation by simultaneously exploiting image and event modalities. arXiv preprint arXiv:2211.10754.

Bohte, S. M., Kok, J. N., & La Poutre, H. (2002). Error-backpropagation in temporally encoded networks of spiking neurons. *Neurocomputing*, *48*(1–4), 17–37.

Brandli, C., Berner, R., Yang, M., Liu, S.-C., & Delbruck, T. (2014). A 240× 180 130 db 3 μs latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits*, *49*(10), 2333–2341.

Brostow, G. J., Fauqueur, J., & Cipolla, R. (2009). Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, *30*(2), 88–97.

Brostow, G. J., Shotton, J., Fauqueur, J., & Cipolla, R. (2008). Segmentation and recognition using structure from motion point clouds. In *Computer vision–ECCV 2008: 10th European conference on computer vision, marseille, France, October 12-18, 2008, proceedings, part i 10* (pp. 44–57). Springer.

Cao, Y., Chen, Y., & Khosla, D. (2015). Spiking deep convolutional neural networks for energy-efficient object recognition. *International Journal of Computer Vision*, *113*, 54–66.

Chen, Z., Deng, L., Wang, B., Li, G., & Xie, Y. (2020). A comprehensive and modularized statistical framework for gradient norm equality in deep neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(1), 13–31.

Chen, H., Wang, Y., Xu, C., Yang, Z., Liu, C., Shi, B., et al. (2019). Data-free learning of student networks. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 3514–3522).

Chen, L. C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision* (pp. 801–818).

Cheni, Q., Rueckauer, B., Li, L., Delbruck, T., & Liu, S. C. (2021). Reducing latency in a converted spiking video segmentation network. In *2021 IEEE international symposium on circuits and systems* (pp. 1–5). IEEE.

Cho, S. J., Ji, S. W., Hong, J. P., Jung, S. W., & Ko, S. J. (2021). Rethinking coarse-to-fine approach in single image deblurring. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 4641–4650).

Comsa, I. M., Potempa, K., Versari, L., Fischbacher, T., Gesmundo, A., & Alakuijala, J. (2020). Temporal coding in spiking neural networks with alpha synaptic function. In *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing* (pp. 8529–8533). IEEE.

Davies, M., Srinivasa, N., Lin, T. H., Chinya, G., Cao, Y., Choday, S. H., et al. (2018). Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro*, *38*(1), 82–99.

Deng, L., Wu, Y., Hu, X., Liang, L., Ding, Y., Li, G., et al. (2020). Rethinking the performance comparison between SNNS and ANNS. *Neural Networks*, *121*, 294–307.

Diehl, P. U., Neil, D., Binas, J., Cook, M., Liu, S. C., & Pfeiffer, M. (2015). Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In *2015 international joint conference on neural networks* (pp. 1–8). IEEE.

Emek Soylu, B., Guzel, M. S., Bostanci, G. E., Ekinci, F., Asuroglu, T., & Acici, K. (2023). Deep-learning-based approaches for semantic segmentation of natural scene images: A review. *Electronics*, *12*(12), 2730.

Fang, W., Yu, Z., Chen, Y., Huang, T., Masquelier, T., & Tian, Y. (2021). Deep residual learning in spiking neural networks. *Advances in Neural Information Processing Systems*, *34*, 21056–21069.

Fang, W., Yu, Z., Chen, Y., Masquelier, T., Huang, T., & Tian, Y. (2021). Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 2661–2671).

Frenkel, C., Legat, J. D., & Bol, D. (2019). A 65-nm 738k-synapse/mm 2 quad-core binary-weight digital neuromorphic processor with stochastic spike-driven online learning. In *2019 IEEE international symposium on circuits and systems* (pp. 1–5). IEEE.

Gehrig, M., Aarents, W., Gehrig, D., & Scaramuzza, D. (2021). Dsec: A stereo event camera dataset for driving scenarios. *IEEE Robotics and Automation Letters*, *6*(3), 4947–4954.

Gehrig, D., Gehrig, M., Hidalgo-Carrió, J., & Scaramuzza, D. (2020). Video to events: Recycling video datasets for event cameras. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3586–3595).

Guo, Y., Liu, Y., Georgiou, T., & Lew, M. S. (2018). A review of semantic segmentation using deep neural networks. *International Journal of Multimedia Information Retrieval*, *7*, 87–93.

Haessig, G., Cassidy, A., Alvarez, R., Benosman, R., & Orchard, G. (2018). Spiking optical flow for event-based sensors using ibm's truenorth neurosynaptic system. *IEEE Transactions on Biomedical Circuits and Systems*, *12*(4), 860–870.

Han, S., Mao, H., & Dally, W. J. (2015). Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. arXiv preprint arXiv:1510.00149.

Han, S., Pool, J., Tran, J., & Dally, W. (2015). Learning both weights and connections for efficient neural network. *Advances in Neural Information Processing Systems*, *28*.

Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., et al. (2018). Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in Neural Information Processing Systems*, *31*.

He, Y., Zhang, X., & Sun, J. (2017). Channel pruning for accelerating very deep neural networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 1389–1397).

Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531.

Hodgkin, A. L., & Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, *117*(4).

Horowitz, M. (2014). 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* (pp. 10–14). IEEE.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., et al. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.

Hu, Y., Liu, S. C., & Delbruck, T. (2021). v2e: From video frames to realistic DVS events. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 1312–1321).

Hu, Y., Wu, Y., Deng, L., & Li, G. (2021). Advancing residual learning towards powerful deep spiking neural networks. arXiv preprint arXiv:2112.08954.

Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. arXiv preprint arXiv:1602.07360.

Izhikevich, E. M. (2003). Simple model of spiking neurons. *IEEE Transactions on Neural Networks*, *14*(6), 1569–1572.

Jia, Z., You, K., He, W., Tian, Y., Feng, Y., Wang, Y., et al. (2023). Event-based semantic segmentation with posterior attention. *IEEE Transactions on Image Processing*, *32*, 1829–1842.

Kamata, H., Mukuta, Y., & Harada, T. (2022). Fully spiking variational autoencoder. In *Proceedings of the AAAI conference on artificial intelligence, vol. 36, no. 6* (pp. 7059–7067).

Kim, Y., Chough, J., & Panda, P. (2022). Beyond classification: Directly training spiking neural networks for semantic segmentation. *Neuromorphic Computing and Engineering*, *2*(4), Article 044015.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., et al. (2023). Segment anything. arXiv preprint arXiv:2304.02643.

Kirkland, P., Di Caterina, G., Soraghan, J., & Matich, G. (2020). Spikeseg: Spiking segmentation via STDP saliency mapping. In *2020 international joint conference on neural networks* (pp. 1–8). IEEE.

Li, G., Deng, L., Tang, H., Pan, G., Tian, Y., Roy, K., et al. (2023). Brain inspired computing: A systematic survey and future trends. TechRxiv.

Li, Y., He, X., Dong, Y., Kong, Q., & Zeng, Y. (2022). Spike calibration: Fast and accurate conversion of spiking neural network for object detection and segmentation. arXiv preprint arXiv:2207.02702.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431–3440).

Ma, N., Zhang, X., Zheng, H. T., & Sun, J. (2018). Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision* (pp. 116–131).

Maass, W. (1997). Networks of spiking neurons: the third generation of neural network models. *Neural Networks*, *10*(9), 1659–1671.

Merolla, P. A., Arthur, J. V., Alvarez-Icaza, R., Cassidy, A. S., Sawada, J., Akopyan, F., et al. (2014). A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, *345*(6197), 668–673.

Messikommer, N., Gehrig, D., Gehrig, M., & Scaramuzza, D. (2022). Bridging the gap between events and frames through unsupervised domain adaptation. *IEEE Robotics and Automation Letters*, *7*(2), 3515–3522.

Neftci, E. O., Mostafa, H., & Zenke, F. (2019). Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, *36*(6), 51–63.

Patel, K., Hunsberger, E., Batir, S., & Eliasmith, C. (2021). A spiking neural network for image segmentation. arXiv preprint arXiv:2106.08921.

Pei, J., Deng, L., et al. (2019). Towards artificial general intelligence with hybrid Tianjic chip architecture. *Nature*, *572*(7767), 106–111.

Posch, C., Matolin, D., & Wohlgenannt, R. (2010). A QVGA 143 dB dynamic range frame-free PWM image sensor with lossless pixel-level video compression and time-domain CDS. *IEEE Journal of Solid-State Circuits*, *46*(1), 259–275.

Qammaz, A., & Argyros, A. A. (2019). MocapNET: Ensemble of SNN encoders for 3D human pose estimation in RGB images. In *BMVC* (p. 46).

Ran, X., Zhang, J., Ye, Z., Wu, H., Xu, Q., Zhou, H., et al. (2021). Deep auto-encoder with neural response. arXiv preprint arXiv:2111.15309.

Rebecq, H., Ranftl, R., Koltun, V., & Scaramuzza, D. (2019). High speed and high dynamic range video with an event camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(6), 1964–1980.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18* (pp. 234–241). Springer.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4510–4520).

Sun, Z., Messikommer, N., Gehrig, D., & Scaramuzza, D. (2022). Ess: Learning event-based semantic segmentation from still images. In *Computer vision–ECCV 2022: 17th European conference, Tel Aviv, Israel, October 23–27, 2022, proceedings, part XXXIV* (pp. 341–357). Springer.

Sun, K., Zhao, Y., Jiang, B., Cheng, T., Xiao, B., Liu, D., et al. (2019). High-resolution representations for labeling pixels and regions. arXiv preprint arXiv:1904.04514.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818–2826).

Wang, L., Chae, Y., & Yoon, K. J. (2021). Dual transfer learning for event-based end-task prediction via pluggable event to image translation. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 2135–2145).

Wang, L., Chae, Y., Yoon, S. H., Kim, T. K., & Yoon, K. J. (2021). Evdistill: Asynchronous events to end-task learning via bidirectional reconstruction-guided cross-modal knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 608–619).

Wu, Y., Deng, L., Li, G., Zhu, J., & Shi, L. (2018). Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in Neuroscience*, *12*, 331.

Wu, Y., Deng, L., Li, G., Zhu, J., Xie, Y., & Shi, L. (2019). Direct training for spiking neural networks: Faster, larger, better. In *Proceedings of the AAAI conference on artificial intelligence, vol. 33, no. 01* (pp. 1311–1318).

Wu, Y., Wang, D. H., Lu, X. T., Yang, F., Yao, M., Dong, W. S., et al. (2022). Efficient visual recognition: A survey on recent advances and brain-inspired methodologies. *Machine Intelligence Research*, *19*(5), 366–411.

Wu, J., Xu, C., Han, X., Zhou, D., Zhang, M., Li, H., et al. (2021). Progressive tandem learning for pattern recognition with deep spiking neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(11), 7824–7840.

Xu, Q., Li, Y., Shen, J., Liu, J. K., Tang, H., & Pan, G. (2023). Constructing deep spiking neural networks from artificial neural networks with knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 7886–7895).

Xu, Q., Li, Y., Shen, J., Zhang, P., Liu, J. K., Tang, H., et al. (2022). Hierarchical spiking-based model for efficient image classification with enhanced feature extraction and encoding. *IEEE Transactions on Neural Networks and Learning Systems*.

Xu, Q., Shen, J., Ran, X., Tang, H., Pan, G., & Liu, J. K. (2021). Robust transcoding sensory information with neural spikes. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(5), 1935–1946.

Yang, J., Deng, L., Yang, Y., Xie, Y., & Li, G. (2021). Training and inference for integer-based semantic segmentation network. *Neurocomputing*, *454*, 101–112.

Yao, M., Hu, J., Zhou, Z., Yuan, L., Tian, Y., Xu, B., et al. (2023). Spike-driven transformer. arXiv preprint arXiv:2307.01694.

Yao, M., Zhao, G., Zhang, H., Hu, Y., Deng, L., Tian, Y., et al. (2023). Attention spiking neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*(8), 9393–9410. http://dx.doi.org/10.1109/TPAMI.2023.3241201.

Zhang, J., Dong, B., Zhang, H., Ding, J., Heide, F., Yin, B., et al. (2022). Spiking transformers for event-based single object tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8801–8810).

Zhang, H., Fan, X., & Zhang, Y. (2023). Energy-efficient spiking segmenter for frame and event-based images. *Biomimetics*, *8*(4), 356.

Zhang, Y., Qu, P., et al. (2020). A system hierarchy for brain-inspired computing. *Nature*, *586*(7829), 378–384.

Zhang, X., Zhou, X., Lin, M., & Sun, J. (2018). Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6848–6856).

Zheng, H., Wu, Y., Deng, L., Hu, Y., & Li, G. (2021). Going deeper with directly-trained larger spiking neural networks. In *Proceedings of the AAAI conference on artificial intelligence, vol. 35, no. 12* (pp. 11062–11070).

Zhu, L., Wang, X., Chang, Y., Li, J., Huang, T., & Tian, Y. (2022). Event-based video reconstruction via potential-assisted spiking neural network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3594–3604).