SFOD: Spiking Fusion Object Detector

Yimeng Fan[1], Wei Zhang[1,2,*], Changsong Liu[1], Mingyang Li[1], Wenrui Lu[1]
[1]School of Microelectronics, Tianjin University, China
[2]Tianjin Key Laboratory of Low-dimensional Electronic Materials and Advanced Instrumentation
{yimengfan, tjuzhangwei, changsong, limingyang97, 3018232176}@tju.edu.cn

Abstract

Event cameras, characterized by high temporal resolution, high dynamic range, low power consumption, and high pixel bandwidth, offer unique capabilities for object detection in specialized contexts. Despite these advantages, the inherent sparsity and asynchrony of event data pose challenges to existing object detection algorithms. Spiking Neural Networks (SNNs), inspired by the way the human brain codes and processes information, offer a potential solution to these difficulties. However, their performance in object detection using event cameras is limited in current implementations. In this paper, we propose the Spiking Fusion Object Detector (SFOD), a simple and efficient approach to SNN-based object detection. Specifically, we design a Spiking Fusion Module, achieving the first-time fusion of feature maps from different scales in SNNs applied to event cameras. Additionally, through integrating our analysis and experiments conducted during the pretraining of the backbone network on the NCAR dataset, we delve deeply into the impact of spiking decoding strategies and loss functions on model performance. Thereby, we establish state-of-the-art classification results based on SNNs, achieving 93.7% accuracy on the NCAR dataset. Experimental results on the GEN1 detection dataset demonstrate that the SFOD achieves a state-of-the-art mAP of 32.1%, outperforming existing SNN-based approaches. Our research not only underscores the potential of SNNs in object detection with event cameras but also propels the advancement of SNNs. Code is available at https://github.com/yimeng-fan/SFOD.

1. Introduction

Event cameras are visual sensors that capture images in a novel manner. In contrast to conventional frame cameras that record complete images at a fixed rate, event cameras asynchronously collect changes in brightness at each pixel.

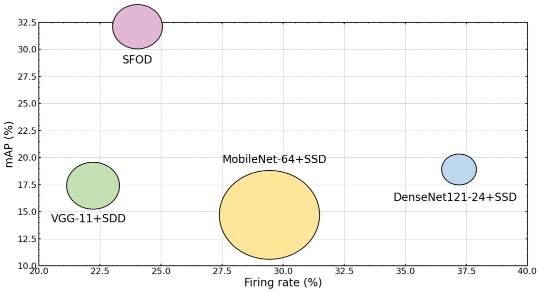

Figure 1. **Detection performance vs firing rate of our SFOD on the GEN1 dataset.** The areas of the circles correspond to the model size.

Consequently, event cameras present noteworthy qualities, including high temporal resolution, high dynamic range, low power consumption, and high pixel bandwidth [13]. These attributes provide them with many advantages in object detection tasks, particularly in scenarios characterized by rapid motion or complex lighting conditions. However, the rapid sampling rate and the sparse, asynchronous format of event data present significant challenges to existing object detection algorithms. Consequently, it is crucial to address these challenges in the domain of event-based object detection research.

Spiking Neural Networks (SNNs), recognized as the third generation of neural networks [29, 36], are considered a promising solution. Unlike non-Spiking Neural Networks (non-SNNs), SNNs emulate the coding and processing of information in the human brain by utilizing spiking neurons as computational units [29]. This makes them inherently suited for processing event data. However, existing research on SNN-based object detection models applied to event cameras remains relatively limited [9]. Among them, the most crucial aspect that has not been thoroughly explored is the fusion of multi-scale feature maps. Such fusion is more important in SNNs than in non-SNNs, as it not only achieves a combination of deeper and shallower feature maps in the spatial domain but also enhances con-

[*]Corresponding author.

nections between features of different scales in the temporal domain. For instance, when recording a person walking with an event camera, the shallow layers of SNNs might initially detect only leg movements. However, as the person completes the movement, the deeper layers capture the full action. Multi-scale fusion in SNNs integrates these varying temporal perceptions from different layers, thereby significantly improving detection accuracy. In contrast, non-SNNs for RGB images only focus on spatial fusion and struggle with temporal data from event cameras, where RNNs can help but need more complexity and computational demands.

To tackle this problem, we propose the Spiking Fusion Module that realizes the fusion of multi-scale feature maps in SNNs applied to event cameras for the first time. This module is further combined with the Spiking DenseNet [9] and the SSD detection head [25] to form the Spiking Fusion Object Detector (SFOD). In essence, our feature fusion strategy is as follows: multi-scale feature maps are extracted from the backbone network, which, after upsampling and concatenation, are fed into the Spiking Pyramid Extraction Submodule (SPES) to further refine the feature representations.

Furthermore, in SNNs, spike trains are used for the coding and processing of information. Therefore, decoding these spike trains by an effective strategy at the output layer is important for accurate inference. However, there is currently no corresponding research on spiking decoding for SNNs applied to event cameras. To investigate this, we perform a comprehensive analysis of different decoding strategies and conduct experiments with them during the pretraining phase of the backbone network. Concurrently, we study different classification loss functions to improve model performance. Our findings indicate that combining Spiking Rate Decoding with Mean Squared Error (MSE) loss function produces the best classification performance. As a result, we achieve state-of-the-art accuracy with SNNs on the NCAR dataset [40]. In further experiments on the detection model, we evaluate the effect of different spiking decoding strategies and confirm that Spiking Rate Decoding can significantly improve the performance of the model.

The main contributions of this work can be summarized as follows:

(1) We propose Spiking Fusion Module, which is the first to implement spiking feature fusion in SNNs for event cameras. It extracts and refines multi-scale feature maps from the backbone network in an SNN-friendly manner. This enhances the model's detection capabilities for targets of various scales. Furthermore, by integrating this module with Spiking DenseNet and SSD detection head, we design the SFOD, a simple and efficient SNN-based object detector.

(2) For the first time in SNNs applied to event cameras, we conduct a thorough study of different spiking decoding strategies and classification loss functions to determine their impact on model performance. On the NCAR dataset, utilizing Spiking Rate Decoding paired with MSE loss, we achieve the state-of-the-art classification result based on SNNs, with an accuracy of 93.7%.

(3) On the GEN1 dataset [10], our SFOD achieves the state-of-the-art object detection performance of 32.1% mAP for SNN-based models. Notably, compared to previous SNN-based detection models, SFOD not only demonstrates a significant enhancement in mAP but also maintains the model parameters and firing rate at a comparable level.

2. Related Work

2.1. Spiking Neural Networks

SNNs can more closely mimic the spiking behavior of biological neurons than non-SNNs. To pursue this biomimicry, scholars have proposed various spiking neural models, including Hodgkin-Huxley (H-H) model [16], Izhikevich model [18], Leaky Integrate-and-Fire (LIF) model [1], and Parametric Leaky Integrate-and-Fire (PLIF) model [12]. Among these, the PLIF model, with its simplicity and capability to reduce the networks' sensitivity to initial conditions, has been widely adopted in current applications.

In SNNs, there are primarily three spiking decoding strategies: Spiking Count Decoding, Spiking Rate Decoding, and Membrane Potential Accumulation Decoding [38]. Specifically, the Spiking Count Decoding counts spikes over a given duration, while the Spiking Rate Decoding divides this count by time T, meanwhile the Membrane Potential Accumulation Decoding prohibits spike firing in the final layer and conveys information by accumulating membrane potentials.

SNNs primarily adopt two training strategies: ANNs-to-SNNs conversion and direct training. The ANNs-to-SNNs conversion uses the spiking rate to simulate the ReLU activation, enabling the transformation of trained ANNs into SNNs [6, 37]. Although this method has enabled the realization of powerful SNNs, such as Spiking YOLO proposed by [20], it requires thousands of time steps and is only suitable for static datasets rather than event-driven datasets. In contrast, the direct training approach employs surrogate gradient methods to optimize SNNs directly [31], allowing SNNs to be trained on various datasets and achieve competitive performance within a few time steps. The success of this strategy has led to the widespread use of SNNs in visual tasks, including image classification [11], object detection [9], and video reconstruction [43], etc. Consequently, we adopt the direct training strategy for our model in this work.

2.2. Object Detection for Event Cameras

For object detection tasks with event cameras, one straightforward approach is to generate images frame-by-frame through temporal integration, thereby facilitating dense op-

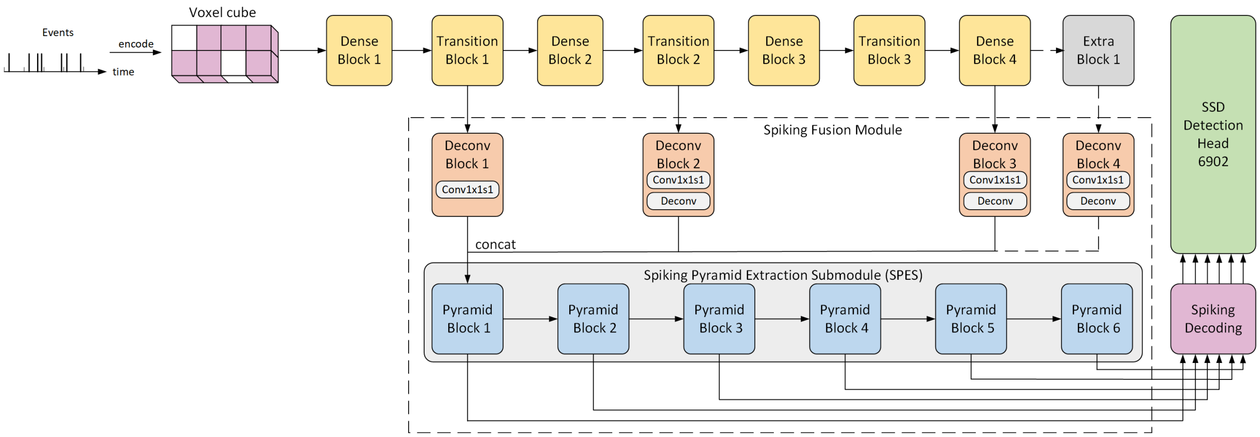

Figure 2. **The architecture of SFOD.** The Spiking Fusion Module is highlighted in the dotted area of the figure. In the fusion of layer four, Extra Block1 and Deconv Block4 are introduced and connected with the remainder of the network through the dotted lines.

erations like convolution. Subsequently, traditional non-SNNs can be trained for object detection. This method has been extensively adopted in studies such as [4, 8, 17, 19]. However, a significant drawback of this method is the loss of temporal information inherent in event data. To address this, algorithms like [5, 15, 32] have incorporated RNNs and Transformers, employing coding strategies that preserve temporal characteristics, thereby enhancing detection performance considerably. However, we believe that these techniques do not take full advantage of the sparsity inherent in event data as SNNs do. Although [9, 41] pioneer the use of SNNs for object detection, the detection performance achieved remains suboptimal. Following these, we aim to further investigate and enhance SNN-based object detection algorithms applied to event cameras.

3. Method

3.1. Overview

The architecture of our proposed SFOD is illustrated in Figure 2. Initially, for the model to process sparse and asynchronous event data, we adopt the voxel cube [9] to code event data. Following this, we provide a brief explanation of the coding method. For event data ϵ within the time interval $[t_a, t_b)$, the voxel cube can be expressed as:

$$E(\tau, c, x, y) = \sum_{e_k \in \epsilon} \delta\left(\tau - \tau_k\right) \delta\left(c - c_k\right) \delta\left(x - x_k, y - y_k\right) \tag{1}$$

$$\tau_k = \left\lfloor \frac{t_k - t_a}{t_a - t_b} \cdot T \right\rfloor \tag{2}$$

Herein, t_k indicates the timestamp of the event occurrence, (x_k, y_k) denotes the pixel coordinates and T represents the number of time bins. To make more effective use of the temporal information contained in events, we further divide each time bin into n micro time bins. Combined with the event polarity $P \in \{0, 1\}$, this results in a total of C channels, where $C = 2n$.

Then, to ensure efficient feature extraction, we employ Spiking DenseNet as the backbone network. This choice is motivated by its outstanding performance in prior work [9]. Subsequently, with the aim of extracting deeper feature maps from the backbone network, we adopt the Extra Block from [9], which is composed of 1x1 and 3x3 convolutional layers. Based on this, the Spiking Fusion Module fuses and enhances the feature maps selected from the backbone network and the Extra Block. Finally, the processed feature maps are fed to the SSD detection head.

Notably, the SSD detection head consists solely of convolutions without any activation functions. Therefore, before feeding the features into it, we perform spiking decoding on these feature maps. Moreover, in order for the network to match the characteristics of SNNs, we replace all traditional activation functions with PLIF neurons.

In the following, we will elucidate the key design elements within the algorithm.

3.2. Spiking Fusion Module

In the current SNNs applied to event camera object detection, there is a lack of studies on the fusion of multi-scale feature maps. To tackle this question, we propose the Spiking Fusion Module, a novel and efficient feature fusion module expressly designed for SNNs. With this method, we achieve the fusion within the spatial and temporal domains of the multi-scale feature maps. This improvement allows the model to better extract features and detect targets at various scales, resulting in a significant enhancement in its performance. The Spiking Fusion Module can be described as follows:

$$X_p = \Phi_p\left(\Phi_f\left(\cup\left\{\Phi_i\left(X_i\right)\right\}\right)\right) \quad i \in \mathcal{I},\ p \in \mathcal{P} \tag{3}$$

In this formula, X_i where $i \in \mathcal{I}$ represents the raw feature maps, while X_p with $p \in \mathcal{P}$ denotes the newly generated feature maps post-fusion. The Φ_i delineates the transformation function for the raw feature maps, Φ_f stands for the feature fusion function and Φ_p is the pyramid feature generation function. The Φ_i, Φ_f and Φ_p respectively correspond to the Deconv Block, concat and SPES in Figure 2.

Next, we will discuss and analyze the design of $\mathcal{I}$, Φ_i, Φ_f and Φ_p in the context of SNNs' characteristics.

$\mathcal{I}$: In the [9], the authors select six feature maps from Spiking DenseNet and its three appended Extra Blocks. These feature maps are then fed into the head network for object detection. The resolutions of these six feature maps are 30x38, 15x19, 7x9, 4x5, 2x3, and 1x2, respectively. We observe that although the deeper feature maps have a larger receptive field and can capture a broader context information, their spatial resolution is inadequate for effective fusion with the shallower layers. Consequently, we discard the last two feature maps (2x3 and 1x2) and conduct experiments to determine whether to retain the fourth feature map (4x5). As shown in Section 4.3, the performance of ignoring the fourth layer surpasses that of using it. Therefore, we opt to fuse the first three layers (30x38, 15x19, and 7x9).

Φ_f: In the non-spiking object detection models, there are primarily two methods for feature fusion. One involves fusing multi-scale feature maps by concatenation, such as ION [2], HyperNet [21], and MFSSD [42]. The other connects different feature maps with element-sum, such as U-Net [35] and FPN [24]. However, in our perspective, employing the element-wise summation approach disrupts the binariness of SNNs, thereby increasing the computational complexity of the algorithm. Consequently, we adopt the method of concatenating multiple feature maps for fusion.

Φ_i: To employ the concatenation method for feature fusion, we upsample feature maps with resolutions smaller than 30x38 to ensure the same spatial dimensions across all feature maps. While bilinear interpolation is a prevalent upsampling technique in traditional frame-based image processing [7, 42], this method involves numerous multiplication and division operations, which might impair the inherent binariness of SNNs. Consequently, we opt for transposed convolution (often referred to as deconvolution) [26] for image upsampling. This approach not only preserves the characteristics of SNNs but also exhibits adaptability to spiking data. Furthermore, prior to the transposed convolution, we use a 1x1 convolution both to refine non-linear feature representations and to ensure the same channel dimensions across all feature maps, allowing the network to allocate equal attention to each feature map.

Φ_p: To regenerate and refine multi-scale feature maps, we propose the Spiking Pyramid Extraction Submodule (SPES). As illustrated in Figure 3, we present three variant SPES architectures. Architecture (a) is the basic version, in which the Pyramid Block is formed by 1x1 and 3x3 convolutions. As proposed by [3], enhancing the head network of an object detection model can effectively boost its performance. However, in our model, the head network operates without spiking. Given this characteristic and the fact that the output of each Pyramid Block is fed into the head network, we aim to indirectly improve the head network's performance by enhancing SPES. Thus, we propose architectures (b) and (c) as enhanced versions of (a). Notably, with our network exceeding 100 layers and to avoid potential gradient vanishing from enhanced SPES, architectures (b) and (c) employ the Spike-Element-Wise Residual Block (SEW Res Block) [11] and Spiking Dense Block [9] for enhancement, respectively. Finally, we adopt the SEW Res Block to enhance SPES for the efficient extraction of multi-scale feature maps, as supported by the experimental results and analysis in Section 4.3.

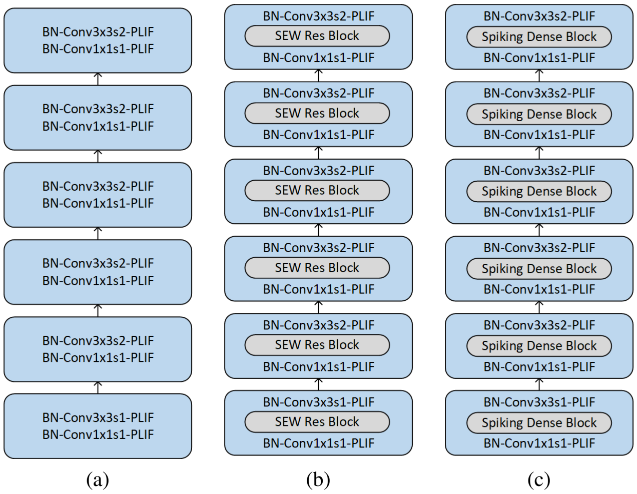

Figure 3. **The architectures of SPES.** The blue block corresponds to the Pyramid Block in Figure 2.

The Spiking Fusion Module can be summarized as follows: First, the Deconv Block conducts 1x1 convolution and transposed convolution upsampling on the first three feature maps. Subsequently, the processed feature maps with the same spatial dimensions are concatenated for feature fusion. Finally, the fused feature maps are fed into the SPES to regenerate multi-scale feature maps.

3.3. Spiking Decoding and Loss Function

As discussed in Section 2.1, there are primarily three spiking decoding strategies: Spiking Count Decoding, Spiking Rate Decoding, and Membrane Potential Accumulation Decoding. We contend that Membrane Potential Accumulation Decoding, by omitting the step of neuronal spike firing, not only disrupts the inherent information processing mechanisms of SNNs but also diminishes their non-linear expression capabilities. Therefore, our research focuses predominantly on Spiking Count Decoding and Spiking Rate Decoding. According to our analysis, the distinction between them lies in the normalization of output in Spiking Rate Decoding. This leads to a more uniform decoding result.

Moreover, when pretraining the backbone network for classification task, our analysis indicates that the Mean Squared Error (MSE) loss is more appropriate than the Cross-Entropy (CE) loss. The reasons are as follows:

First, for the discrete spike counts or frequencies decoded by the model, MSE can compute the loss and gradients directly. However, CE requires an extra softmax step to convert these discrete values into probability distributions, increasing the computational complexity. Second, the gradients of MSE directly represent the difference between decoded values and the labels, while the gradients of CE reflect the difference between post-softmax probability values and the labels. This characteristic of CE is suitable for non-SNNs, as the continuous floating-point outputs of these networks, when processed via softmax, can closely approximate the labels. However, SNNs produce limited discrete output values. This leads to optimization issues in the neurons when using CE loss, consequently reducing the model's generalization ability and increasing its firing rate. For example, when the model employs Spiking Rate Decoding for binary classification and assumes label of $[0, 1]$, consider two cases with decoded outputs of $[0.2, 0.8]$ and $[0.2, 1.0]$. Using CE, the results of the softmax computations are $[0.35, 0.65]$ and $[0.31, 0.69]$, respectively. Even though the decoded value for the negative class remains constant, change in probability values leads to reduced gradients for the negative class neuron. Meanwhile, the gradients for the positive class neuron, while decreased, still remain. In comparison, MSE derives the gradients directly from the decoded output, ensuring consistent gradients for the negative class neuron. When the decoded value for the positive class reaches 1.0, its gradients reduce to zero, halting further optimization.

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C}(y_{ij} - a_{ij})^2 \tag{4}$$

$$\text{CE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\log(z_{ij}) \tag{5}$$

The formulas for MSE and CE are described in Equations 4 and 5, respectively. In these equations, N is the sample size, C is the number of classes, while y_{ij}, a_{ij} and z_{ij} represent the label, decoded value, and post-softmax probability value of the j^{th} class for the i^{th} sample, respectively.

Building on the above analysis, we conduct experiments on the NCAR dataset evaluating all combinations of decoding strategies and loss functions, as elaborated in Section 4.2. It's noteworthy that the combination of Spiking Rate Decoding and the MSE loss yields the best classification results. Additionally, in object detection task, experiments also show that Spiking Rate Decoding outperforms Spiking Count Decoding. Consequently, we opt for Spiking Rate Decoding as the decoding strategy in SFOD.

Models	Dec.	Loss Func.	Params	Acc.	Firing Rate
DenseNet 121-16	**Rate**	**MSE**	**1.76M**	**0.937**	**14.70%**
DenseNet 121-16	Count	MSE	1.76M	0.869	12.95%
DenseNet 121-16	Rate	CE	1.76M	0.930	20.42%
DenseNet 121-16	Count	CE	1.76M	0.920	17.09%

Table 1. **Spiking decoding and loss functions.** Comparison of different spiking decoding strategies and loss functions on the NCAR dataset.

4. Experiment

In this Section, we first investigate the effects of different spiking decoding strategies and loss functions, and then pretrain the backbone networks, both on the NCAR dataset. Subsequently, we conduct an ablation study on the GEN1 dataset for SFOD and compare the best-performing model with state-of-the-art methods.

4.1. Experiment Setup

Datasets. The NCAR dataset [40] is a binary classification dataset, comprising 12,336 car samples and 11,693 background samples. Each sample has a duration of 100 ms and exhibits varying spatial dimensions.

The GEN1 dataset [10] is the first large-scale object detection dataset captured by event camera. It consists of over 39 hours of car videos recorded by the GEN1 camera. Bounding box labels for cars and pedestrians within the recordings are provided at frequencies between 1 to 4Hz, amassing over 255,000 labels in total.

Implementation Details. We code all samples with voxel cube, using a T-value of 5 and a micro time bin of 2, as done in [9]. All models are trained using the AdamW optimizer [28], coupled with cosine learning rate scheduler [27]. On the NCAR dataset, our models are trained for 30 epochs, with a batch size of 64, an initial learning rate of 5e-3, and a weight decay of 1e-2. On the GEN1 dataset, the training parameters include 50 epochs, a batch size of 16, an initial learning rate of 1e-3, and a weight decay of 1e-4. To increase diversity and balance the sample distribution across

Models	Params	Acc.	Firing Rate
DenseNet121-16	**1.76M**	**0.937**	**14.70%**
DenseNet121-24	3.93M	0.928	15.90%
DenseNet121-32	6.95M	0.923	24.87%
DenseNet169-16	3.16M	0.921	15.34%
DenseNet169-24	7.05M	0.923	19.70%
DenseNet169-32	12.48M	0.894	27.28%

Table 2. **Spiking DenseNets architectures.** Classification performance of different Spiking DenseNets architectures on the NCAR dataset

classes on the GEN1 dataset, we apply horizontal flipping as a data augmentation strategy.

Performance Metrics. For classification tasks, accuracy is the primary evaluation metric. For object detection tasks, the main evaluation metrics are mAP@0.5:0.95 and mAP@0.5. Another crucial metric for evaluating SNNs is the firing rate. It is defined as the average ratio of the number of neuron spikes to the total number of neurons across all time steps, representing the level of neuronal activity. On certain specialized hardware, computations occur only when spikes are emitted. As a result, SNNs with a lower firing rate can notably reduce power consumption.

4.2. Analysis and Pretraining on NCAR Dataset

In this section, we first explore the impact of different spiking decoding strategies and loss functions on model performance. Then, using the optimal combination, we train Spiking DenseNets of various depths and growth rates to study their structural influence. Based on the results, we choose the best-performing models as the backbone networks for our detection models.

Analysis of Spiking Decoding and Loss Function. According to the experimental results in Table 1, the model using Spiking Rate Decoding has better accuracy than the model using Spiking Count Decoding at the same level of firing rate, regardless of the loss function used. Specifically, when utilizing the MSE loss, the model with Spiking Count Decoding exhibits an approximate 7% decline in accuracy compared to that using Spiking Rate Decoding. Table 4 presents the results for object detection models, and rows 2 and 3 further confirm this conclusion. We believe that the primary reason for this difference lies in the consistent setting of the prediction range between $[0, 1]$ for both classification and detection tasks. This makes Spiking Rate Decoding more suitable than Spiking Count Decoding, which has an output range of $[0, T]$ that does not match the prediction range. This mismatch could impact the model's learning efficiency, as it necessitates adjustments within a wider output range. Conversely, the output range of Spiking Rate Decoding is normalized to align with the prediction values, thereby more effectively reflecting prediction errors and enhancing the model's learning ability.

Methods	Networks	Acc.	Firing Rate
HATS [40]	N/A	0.902	-
HybridSNN [22]	SNNs-CNNs	0.906	-
YOLOE [4]	CNNs	0.927	-
EvS-S [23]	GNNs	0.931	-
Asynet [30]	**CNNs**	**0.944**	**-**
HybridSNN [22]	SNNs	0.770	-
Gabor-SNN [40]	SNNs	0.789	-
SqueezeNet 1.1 [9]	SNNs	0.846	25.13%
MobileNet-64 [9]	SNNs	0.917	17.14%
DenseNet169-16 [9]	SNNs	0.904	33.59%
VGG-11 [9]	**SNNs**	**0.924**	**14.69%**
DenseNet121-16	**SNNs**	**0.937**	**14.70%**

Table 3. **Comparison with state-of-the-art models on the NCARS dataset.**

Furthermore, the MSE loss outperforms the CE loss when Spiking Rate Decoding is used, both in terms of accuracy and firing rate. This further substantiates the viewpoint we present in Section 3.3. However, when using Spiking Count Decoding, the MSE does not perform as well as CE. We believe that the transformation of decoded values into probability distributions during the softmax step of CE computation serves as a normalization, which reduces the effects of the un-normalized Spiking Count Decoding. On the other hand, the MSE is extremely sensitive to the deviation between the predicted values and the labels, so using the un-normalized Spiking Count Decoding could lead to a higher accumulation of errors, ultimately affecting the overall performance of the model.

Pretraining on NCAR Dataset. Based on the experimental results above, we further investigate the impact of different architectures of Spiking DenseNets on performance. As shown in Table 2, it can be observed that when the growth rate is fixed, an increase in the model depth leads to a rise in firing rate and a decline in accuracy. Similarly, with a fixed model depth, an increase in the growth rate raises the firing rate, while only slightly reducing the accuracy. Notably, for models with a depth of 169, the one with a growth rate of 24 outperforms the one with a growth rate of 16 in accuracy. Taking into account the trade-off between accuracy and firing rate, we select DenseNet121-16, DenseNet121-24, and DenseNet169-16 as the backbone networks for the detection models, thereby delving deeper into the influence of model architecture on detection performance.

Models	Dec.		Fusion Layers			Params	mAP@0.5:0.95	mAP@0.5	Firing Rate
	Rate	Count	None	3	4				
DenseNet121-16-SSD	✓		✓			5.0M	0.262	0.517	21.01%
DenseNet121_24-SSD		✓	✓			8.2M	0.235	0.445	22.02%
DenseNet121-24-SSD	✓		✓			8.2M	0.288	0.553	22.29%
DenseNet169-16-SSD	✓		✓			7.7M	0.257	0.507	22.82%
SFOD-B	✓				✓	15.0M	0.294	0.570	21.13%
SFOD-B	✓			✓		9.9M	0.299	0.575	24.41%
SFOD-D	✓			✓		11.3M	0.286	0.558	26.37%
SFOD-R	✓			✓		**11.9M**	**0.321**	**0.593**	**24.04%**

Table 4. **Results of the ablation study on the GEN1 dataset.**

Method	Networks	Detection Head	Params	mAP @0.5:0.95	Firing Rate	Time (ms)	Energy (mJ)
Asynet [30]	Sparse CNNs	YOLOv1 [34]	11.4M	0.145	-	-	> 4.83
AEGNN [39]	GNNs	YOLOv1	20.0M	0.163	-	-	-
Inception+SSD [17]	CNNs	SSD [25]	-	0.301	-	19.4	-
MatrixLSTM [5]	RNNs+CNNs	YOLOv3 [33]	61.5M	0.310	-	-	-
RED [32]	RNNs+CNNs	SSD	24.1M	0.400	-	16.7	> 24.08
RVT [15]	**Transformer+RNNs**	**YOLOX [14]**	**18.5M**	**0.472**	**-**	**10.2**	**-**
MobileNet-64+SSD [9]	SNNs	SSD	24.3M	0.147	29.44%	1.7†	5.76
VGG-11+SDD [9]	SNNs	SSD	12.6M	0.174	22.22%	4.4†	11.06
DenseNet121-24+SSD [9]	SNNs	SSD	8.2M	0.189	37.20%	4.1†	3.89
EMS-YOLO [41]	**SNNs**	**YOLOv3**	**14.4M**	**0.310**	**17.80%**	**-**	**-**
SFOD	**SNNs**	**SSD**	**11.9M**	**0.321**	**24.04%**	**6.7**	**7.26**

Table 5. **Comparison with state-of-the-art models on the GEN1 dataset.** A † indicates that the runtime is not directly available and is estimated in the local environment. The method for calculating energy consumption can be found in the supplementary material.

Table 3 presents a comparison between our best-performing model and other state-of-the-art methods on the NCAR dataset. The results indicate that our model not only outperforms other SNN-based methods but also surpasses the majority of methods based on non-SNNs in terms of accuracy. The accuracy gap between our model and the best non-spiking model is only 0.7%.

4.3. Ablation Study on GEN1 Dataset

In this section, we first study the performance differences across object detection models using various backbone networks. Based on this, we select the best backbone and further analyze the impact of different fusion layers. Finally, we compare the performance of various SPES variants. We name the models using the basic, Spiking Dense Block-enhanced, and SEW Res Block-enhanced SPESs as SFOD-B, SFOD-D, and SFOD-R, respectively.

Different Backbone Network Architectures. In Table 4, specifically in rows 1, 3, and 4, we examine the effect of different backbone network architectures on model performance. Importantly, we don't incorporate the proposed Spiking Fusion Module into the detection models for this study. This decision not only leaves our experimental objectives unaffected but also provides a baseline for further experiments. The results indicate that when employing a consistent growth rate, an increase in network depth leads to a decline in mAP. However, at the same depth, a higher growth rate yields improved mAP. Furthermore, based on our observations, variations in network architectures appear to have no discernible impact on the firing rate. Therefore, we utilize DenseNet121-24 as the backbone network.

The Range of Fusion Layers. As demonstrated in rows 5 and 6 of Table 4, although fusing the fourth layer of feature maps effectively reduces firing rate, it does not lead to an improvement in mAP. In light of comprehensive consideration, we prioritize the improvement of mAP over the reduction of firing rate. Therefore, we conclude that a strategy of fusing three layers is superior to fusing four layers.

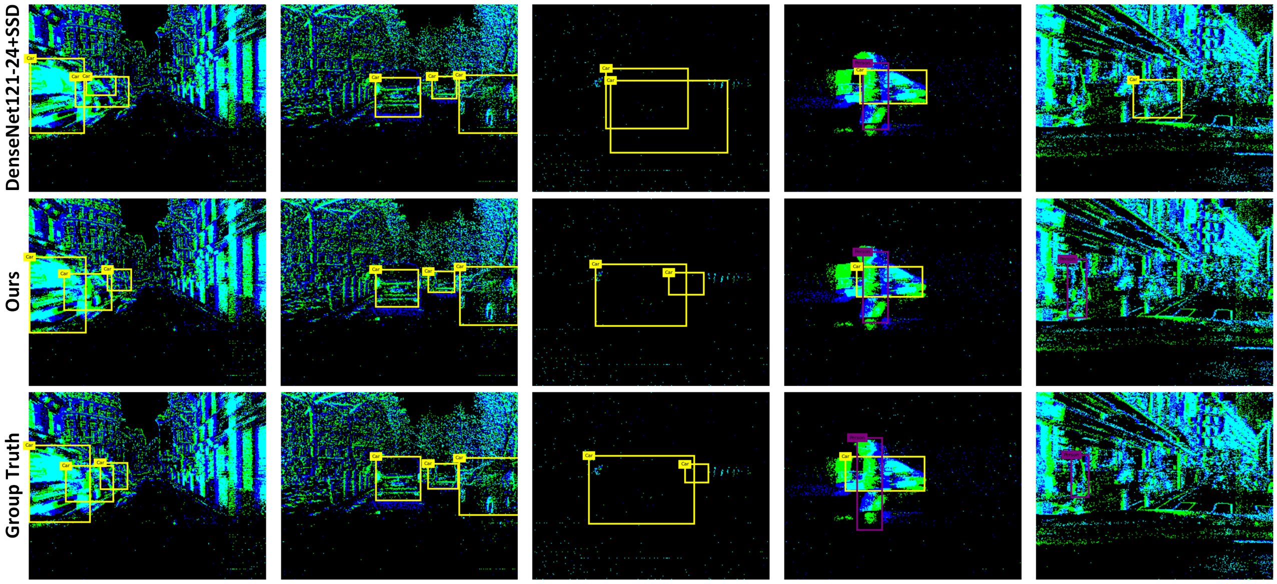

Figure 4. **Inference results of the model on the GEN1 dataset.** The figure illustrates the detection capabilities of the models across specific scenarios: The first column demonstrates detection of overlapping cars; the second showcases non-overlapping detection; the third presents detection in sparse data contexts; the fourth reveals performance in multi-category scenes; and the fifth focuses on individual person target detection.

Comparison of Different SPESs. Rows 6, 7, and 8 of Table 4 present a comparison of various SPES variants. The results reveal that, compared to the SFOD-B, the SFOD-D worsens both the firing rate and the mAP, while the SFOD-R not only shows a slight decrease in firing rate but also significantly improves the mAP by 2.2 points. This suggests that the identity mapping introduced by SEW Res Block can notably elevate model performance, whereas the multi-feature map connection mechanism introduced by Spiking Dense Block results in performance decline. Furthermore, while the integration of the SEW Res Block brings in certain non-spiking computations, these operations, which are limited to the six layers of 3x3 convolutions within SPES, rarely occur in our experiments. Therefore, we think these extra computations are acceptable compared to the significant improvements they bring to SPES.

4.4. Benchmark Comparisons

In Table 5, we present a comparison of our model with other state-of-the-art approaches on the GEN1 dataset. Remarkably, our model achieves a state-of-the-art mAP of 32.1% at the same level of firing rate and parameters compared to other SNN-based methods. This performance nearly doubles that reported in [9]. Additionally, our model surpasses the majority of methods based on non-SNNs. When compared to RED [32] and RVT [15], our model not only has fewer parameters but also demonstrates significant advantages in both energy consumption and computation speed.

Figure 4 presents the results of our model in comparison with DenseNet121-24+SSD [9] and the Ground Truth. The DenseNet121-24+SSD [9] is a reproduced version in the local environment to ensure a fair comparison. From the figure, it is evident that our model consistently outperforms the other model in various scenarios.

5. Conclusion

In this paper, we propose a simple and efficient Spiking Fusion Module. Through this novel approach, we not only establish the current state-of-the-art SNN-based event camera object detection model, SFOD, but also achieve an impressive mAP performance on the GEN1 dataset. Compared to the method reported in [9], the performance of SFOD is nearly double, representing a considerable advancement. Furthermore, during the pretraining phase of the backbone networks, we conduct an in-depth exploration of various combinations of spiking decoding strategies and loss functions. By adopting a combination of Spiking Rate Decoding and MSE, we establish the state-of-the-art SNN-based classification results on the NCAR dataset. More importantly, Spiking Rate Decoding has also significantly contributed to the enhancement in the performance of SFOD.

In the future, we believe that the performance of SFOD is expected to be further improved by adopting a more effective data augmentation strategy. It undeniably represents a promising research direction.

References

[1] Larry F Abbott. Lapicque's introduction of the integrate-and-fire model neuron (1907). *Brain research bulletin*, 50(5-6): 303–304, 1999. 2

[2] Sean Bell, C Lawrence Zitnick, Kavita Bala, and Ross Girshick. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2874–2883, 2016. 4

[3] Zhaowei Cai, Quanfu Fan, Rogerio S Feris, and Nuno Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 354–370. Springer, 2016. 4

[4] Marco Cannici, Marco Ciccone, Andrea Romanoni, and Matteo Matteucci. Asynchronous convolutional networks for object detection in neuromorphic cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 3, 6

[5] Marco Cannici, Marco Ciccone, Andrea Romanoni, and Matteo Matteucci. A differentiable recurrent surface for asynchronous event-based data. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 136–152. Springer, 2020. 3, 7

[6] Yongqiang Cao, Yang Chen, and Deepak Khosla. Spiking deep convolutional neural networks for energy-efficient object recognition. *International Journal of Computer Vision*, 113:54–66, 2015. 2

[7] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014. 4

[8] Nicholas FY Chen. Pseudo-labels for supervised learning on dynamic vision sensor data, applied to object detection under ego-motion. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 644–653, 2018. 3

[9] Loïc Cordone, Benoît Miramond, and Philippe Thierion. Object detection with spiking neural networks on automotive event data. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[10] Pierre de Tournemire, Davide Nitti, Etienne Perot, Davide Migliore, and Amos Sironi. A large scale event-based detection dataset for automotive. 2020. 2, 5

[11] Wei Fang, Zhaofei Yu, Yanqi Chen, Tiejun Huang, Timothée Masquelier, and Yonghong Tian. Deep residual learning in spiking neural networks. *Advances in Neural Information Processing Systems*, 34:21056–21069, 2021. 2, 4

[12] Wei Fang, Zhaofei Yu, Yanqi Chen, Timothée Masquelier, Tiejun Huang, and Yonghong Tian. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2661–2671, 2021. 2

[13] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(1):154–180, 2020. 1

[14] Zheng Ge, Songtao Liu, and Feng Wang Zeming Li Jian Sun. Yolox: Exceeding yolo series in 2021. 7

[15] Mathias Gehrig and Davide Scaramuzza. Recurrent vision transformers for object detection with event cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13884–13893, 2023. 3, 7, 8

[16] Alan L Hodgkin and Andrew F Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117 (4):500, 1952. 2

[17] Massimiliano Iacono, Stefan Weber, Arren Glover, and Chiara Bartolozzi. Towards event-driven object detection with off-the-shelf deep learning. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–9. IEEE, 2018. 3, 7

[18] Eugene M Izhikevich. Simple model of spiking neurons. *IEEE Transactions on neural networks*, 14(6):1569–1572, 2003. 2

[19] Zhuangyi Jiang, Pengfei Xia, Kai Huang, Walter Stechele, Guang Chen, Zhenshan Bing, and Alois Knoll. Mixed frame-/event-driven fast pedestrian detection. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8332–8338. IEEE, 2019. 3

[20] Seijoon Kim, Seongsik Park, Byunggook Na, and Sungroh Yoon. Spiking-yolo: spiking neural network for energy-efficient object detection. In *Proceedings of the AAAI conference on artificial intelligence*, pages 11270–11277, 2020. 2

[21] Tao Kong, Anbang Yao, Yurong Chen, and Fuchun Sun. Hypernet: Towards accurate region proposal generation and joint object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 845–853, 2016. 4

[22] Alexander Kugele, Thomas Pfeil, Michael Pfeiffer, and Elisabetta Chicca. Hybrid snn-ann: Energy-efficient classification and object detection for event-based vision. In *DAGM German Conference on Pattern Recognition*, pages 297–312. Springer, 2021. 6

[23] Yijin Li, Han Zhou, Bangbang Yang, Ye Zhang, Zhaopeng Cui, Hujun Bao, and Guofeng Zhang. Graph-based asynchronous event processing for rapid object recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 934–943, 2021. 6

[24] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 4

[25] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amster-*

dam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14, pages 21–37. Springer, 2016. 2, 7

[26] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 4

[27] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2016. 5

[28] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018. 5

[29] Wolfgang Maass. Networks of spiking neurons: the third generation of neural network models. *Neural networks*, 10 (9):1659–1671, 1997. 1

[30] Nico Messikommer, Daniel Gehrig, Antonio Loquercio, and Davide Scaramuzza. Event-based asynchronous sparse convolutional networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16*, pages 415–431. Springer, 2020. 6, 7

[31] Emre O Neftci, Hesham Mostafa, and Friedemann Zenke. 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, 2019. 2

[32] Etienne Perot, Pierre De Tournemire, Davide Nitti, Jonathan Masci, and Amos Sironi. Learning to detect objects with a 1 megapixel event camera. *Advances in Neural Information Processing Systems*, 33:16639–16652, 2020. 3, 7, 8

[33] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. 7

[34] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 7

[35] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 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*, pages 234–241. Springer, 2015. 4

[36] Kaushik Roy, Akhilesh Jaiswal, and Priyadarshini Panda. Towards spike-based machine intelligence with neuromorphic computing. *Nature*, 575(7784):607–617, 2019. 1

[37] Bodo Rueckauer, Iulia-Alexandra Lungu, Yuhuang Hu, Michael Pfeiffer, and Shih-Chii Liu. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Frontiers in neuroscience*, 11:682, 2017. 2

[38] Bodo Rueckauer, Iulia-Alexandra Lungu, Yuhuang Hu, Michael Pfeiffer, and Shih-Chii Liu. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Frontiers in neuroscience*, 11:682, 2017. 2

[39] Simon Schaefer, Daniel Gehrig, and Davide Scaramuzza. Aegnn: Asynchronous event-based graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12371–12381, 2022. 7

[40] Amos Sironi, Manuele Brambilla, Nicolas Bourdis, Xavier Lagorce, and Ryad Benosman. Hats: Histograms of averaged time surfaces for robust event-based object classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1731–1740, 2018. 2, 5, 6

[41] Qiaoyi Su, Yuhong Chou, Yifan Hu, Jianing Li, Shijie Mei, Ziyang Zhang, and Guoqi Li. Deep directly-trained spiking neural networks for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6555–6565, 2023. 3, 7

[42] Qingcai Wang, Hao Zhang, Xianggong Hong, and Qinqin Zhou. Small object detection based on modified fssd and model compression. In *2021 IEEE 6th International Conference on Signal and Image Processing (ICSIP)*, pages 88–92. IEEE, 2021. 4

[43] Lin Zhu, Xiao Wang, Yi Chang, Jianing Li, Tiejun Huang, and Yonghong Tian. Event-based video reconstruction via potential-assisted spiking neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3594–3604, 2022. 2