Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

S3Det: a fast object detector for remote sensing images based on artificial to spiking neural network conversion*#

Li CHEN[1], Fan ZHANG†‡[1], Guangwei XIE[2], Yanzhao GAO[1], Xiaofeng QI[1], Mingqian SUN[3]

[1]*National Digital Switching System Engineering & Technological R&D Center, Zhengzhou 450003, China*

[2]*Computation and Artificial Intelligence Innovative College, Fudan University, Shanghai 201203, China*

[3]*School of Cyber Science and Engineering, Southeast University, Nanjing 211189, China*

†E-mail: zhangfanryan@163.com

Received July 14, 2024; Revision accepted Oct. 11, 2024; Crosschecked Apr. 29, 2025

Abstract: Artificial neural networks (ANNs) have made great strides in the field of remote sensing image object detection. However, low detection efficiency and high power consumption have always been significant bottlenecks in remote sensing. Spiking neural networks (SNNs) process information in the form of sparse spikes, creating the advantage of high energy efficiency for computer vision tasks. However, most studies have focused on simple classification tasks, and only a few researchers have applied SNNs to object detection in natural images. In this study, we consider the parsimonious nature of biological brains and propose a fast ANN-to-SNN conversion method for remote sensing image detection. We establish a fast sparse model for pulse sequence perception based on group sparse features and conduct transform-domain sparse resampling of the original images to enable fast perception of image features and encoded pulse sequences. In addition, to meet accuracy requirements in relevant remote sensing scenarios, we theoretically analyze the transformation error and propose channel self-decaying weighted normalization (CSWN) to eliminate neuron overactivation. We propose S3Det, a remote sensing image object detection model. Our experiments, based on a large publicly available remote sensing dataset, show that S3Det achieves an accuracy performance similar to that of the ANN. Meanwhile, our transformed network is only 24.32% as sparse as the benchmark and consumes only 1.46 W, which is 1/122 of the original algorithm's power consumption.

Key words: Remote sensing image; Object detection; Spiking neural networks (SNNs); Spiking sequence rapid sensing (SSRS); Channel self-decaying weighted normalization (CSWN)

https://doi.org/10.1631/FITEE.2400594 **CLC number:** TP391

1 Introduction

Due to the rapid development of remote sensing technology in recent years, the automatic analysis of massive remote sensing images with all-round intelligent technology has become urgently needed by academia and industry. Object detection, a fundamental task in this domain, finds extensive application in essential areas such as meteorological analysis, surface surveying, and transportation planning (Chen L et al., 2023).

Artificial neural networks (ANNs), represented by convolutional neural networks (CNNs) (LeCun et al., 1998) and Transformers (Vaswani et al., 2017), have developed rapidly over the past decade and continue to energize object detection tasks. However, ANNs often demand substantial computing resources, posing challenges for their deployment on resource-limited devices. In contrast, spiking neural networks

‡ Corresponding author

* Project supported by the National Key Research and Development Program of China (No. 2022YFB4500900)

Electronic supplementary materials: The online version of this article (https://doi.org/10.1631/FITEE.2400594) contains supplementary materials, which are available to authorized users

ORCID: Li CHEN, https://orcid.org/0009-0006-8206-5255; Fan ZHANG, https://orcid.org/0000-0001-7456-8377

(SNNs) (Maass, 1997) emulate the biological structure of the brain, encoding and transmitting information through $0-1$ spikes, mimicking human biological systems. Because spiking neurons trigger calculations in response to only external stimuli, they have advantages of low power consumption and fast reasoning. At present, SNNs have achieved excellent performance on a variety of hardware platforms, which also validates the feasibility of exploring high-performance and highly efficient brain-inspired intelligence through SNNs.

Despite binary spiking enabling the extreme energy efficiency of SNNs, the complex dynamics involved and the non-differentiable mathematical characteristics of spiking neurons result in a scarcity of training algorithms. Therefore, researchers have tried to find the balance of the scales in the conversion method. ANN-to-SNN conversion (Kim et al., 2020; Li et al., 2022; Hu et al., 2023; Yao et al., 2023) has already yielded promising results on common object detection datasets, such as PASCAL VOC (Everingham et al., 2010) and COCO (Lin et al., 2014). However, as the number of network layers increases, the accuracy error caused by the conversion method further increases. While increasing the firing rate of discrete spikes can significantly mitigate these accuracy errors, it leads to a substantial increase in the time step, undermining the efficiency advantage of SNNs.

To fully harness SNNs' efficiency advantage in real-time remote sensing detection scenarios, our research draws on the principles of simplicity and sparsity inherent in biological information transmission. The simplicity of SNNs lies in their efficient coding, local computation, and event-driven nature. SNNs employ a simple encoding scheme to represent complex information, with neurons communicating primarily with their immediate neighbors, thereby reducing the energy consumption associated with long-distance signal transmission. The event-driven characteristic ensures that neurons generate spikes only when there is a change in the input, leading to a more energy-efficient operation. The sparsity of SNNs is manifested in three key aspects: sparse activation, sparse connectivity, and sparse representation. Sparse activation refers to only a small fraction of neurons being active and generating spikes at any given time. This reduces the overall computational load and energy consumption. Sparse connectivity implies that the connections within the neural network are not fully dense; instead, each neuron typically connects to only a subset of other neurons. Sparse representation, on the other hand, involves using the minimum number of active neurons to encode information.

Guided by the aforementioned principles, we incorporate the theory of compressive sensing from signal processing. Compressive sensing allows for the recovery of sparse signals from a small number of measurements. In the context of SNNs, this can be leveraged to reconstruct complete signals from sparse activation patterns, thereby reducing the number of required sensors and the amount of data. We develop a fast SNN model suitable for remote sensing applications, achieving performance competitive with that of ANNs. The fundamental concept is to eliminate redundant computations and leverage the temporal and spatial features more effectively. In addition, we analyze the causes of errors in the conversion process and make corresponding improvements. The following summarizes the contributions we have made in this work and highlights the benefits of the lightweight SNN that has been proposed:

1. Spiking sequence rapid sensing (SSRS). We analyze the sparse nature of spiking sequences in aerial images and model them using group sparse compressed sensing. Theoretically, we demonstrate that the proposed model effectively performs compressed sampling of the original spiking sequences. This is achieved through the minimization of a mixed norm model.

2. Channel self-decaying weighted normalization (CSWN). To address the issue of excessive activation of spiking neurons, we conduct an in-depth analysis of normalization errors during the conversion from ANN to SNN. Our findings indicate that the spike of inactivated neurons (SIN) error is exacerbated as the number of layers and channels increases. This issue is particularly critical for neural networks that process remote sensing images and must be carefully considered. We recommend an exponentiated momentum decay scheme based on low-order statistics along the channel dimension, which offers a cost-effective solution to this problem.

3. Depth model of remote sensing object detection. We develop SNN models for object detection in remote sensing images, achieving advanced performance and

efficient detection on major publicly available datasets. To the best of our knowledge, this is the first attempt to apply SNNs to object detection tasks in the field of remote sensing.

2 Related works

Here we introduce the remote sensing image object detection methods in the literature. Unlike natural images, remote sensing images are captured from an overhead perspective and include a diverse array of objects such as vehicles, bridges, and ships. These objects are characterized by arbitrary orientations, relatively small and dense objects, and significant scale variations among objects. To address these, researchers have focused on three areas—feature refinement, anchor-free mechanisms, and optimization of the loss function—to develop more accurate object detection algorithms for high-resolution remote sensing images.

In terms of feature refinement, it is necessary to effectively augment rotated image data to maintain rotational invariance. Cheng et al. (2016a) proposed a rotation-invariant CNN (RICNN) that enhances model generalization by using sample rotations. Based on the original model, Cheng et al. (2016b) further incorporated a Fisher discriminant layer to enhance classification similarity, thereby improving classification performance. Additionally, various methods have been employed to refine features through data augmentation, including quad-patch augmentation (Gong et al., 2022), dual-dimensional feature enhancement using multiscale attention CycleGAN (Liu WX et al., 2022), and specifically synthetic mineral oversampling with mosaic and mixup (SSMup) (Chen GH et al., 2022), among others. Enhancing the semantic features of objects is also a common optimization method. Yang et al. (2021a) proposed a progressive regression method called R3Det, which increases accuracy by refining the center points of objects from coarse to fine granularity.

The two-stage detectors with the above-detailed features are all designed based on anchors. However, due to the enumeration and parameter adjustment difficulty of anchors, researchers have gradually expanded the anchor-free algorithm. He et al. (2022) designed HRPNet, which converts the detection of oriented bounding boxes (OBBs) (Fig. 1b) into the regression of an angle and four radii in polar coordinates. This significantly reduced the computational complexity of the network model. Zhang C et al. (2023) used a coarse location module to quickly produce coarsely oriented boxes and then employed a region-based CNN (R-CNN) to complete the detection and classification of objects. The above models eliminate the need for laborious manual anchor design, offering competitive detection speed. Inspired by the characteristics of densely distributed remote sensing objects, Xie et al. (2024a) proposed a method called the objectness activation network (OAN), which predicts whether each image patch contains an object, thereby enhancing detection efficiency. Additionally, Xie et al. (2023) introduced the concept of collaborative learning to address the issues of non-universality in target features and the limitations of single-regression approaches. However, CNNs often suffer from significant feature loss during forward propagation, leading to missed detections and false positives. Consequently, some researchers have opted to optimize the reward and penalty mechanisms directly within the design of the loss function. In the field of remote sensing, the main challenges in designing loss functions are boundary discontinuity and inconsistency with the final detection metrics. To address the boundary discontinuity issue, Yang et al. (2021b) used the Gaussian Wasserstein distance (GWD) to approximate the intersection over union (IoU) rotational loss. Building on Yang's work, Huang et al. (2022) designed a novel general Gaussian heatmap label assignment (GGHL), which not only creates two-dimensional (2D) Gaussian heatmaps but also employs an anchor-free adaptive label assignment strategy to improve detection efficiency.

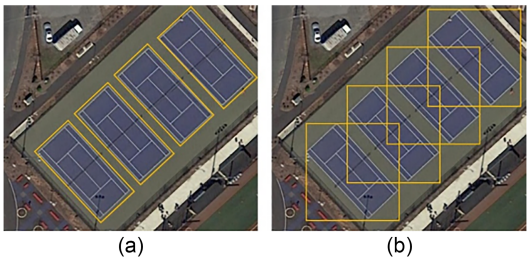

Fig. 1 Two forms of remote sensing image detection boxes. The horizontal bounding box (HBB) (a) contains excessive noise compared with the oriented bounding box (OBB) (b)

All the above methods seek data and network-level optimizations within the framework of ANNs, inherently limiting the upper bound of efficiency. Even when anchor-free single-stage detection algorithms are employed, the resulting detection performance fails to meet the lightweight and low-latency requirements of the remote sensing domain. SNNs transmit information to downstream neurons using action potentials rather than the real values used in ANNs. Due to the sparsity of image inputs, SNNs exhibit significant efficiency advantages.

3 S3Det

3.1 Network architecture

Related work on the conversion of ANNs to SNNs and their proof of equivalence can be found in the supplementary materials. The overall network structure of S3Det is shown in Fig. 2. First, the image is preprocessed with conventional operations, including cropping, scaling, and rotation. The ANN algorithm used for conversion is R3Det. The conversion includes parsing the ANN network, parameter normalization, neuron conversion, and post-processing. In the converted S3Det, we design an SSRS module and CSWN based on low-order statistics.

3.2 Spiking sequence rapid sensing module

In SNNs, the rate-based encoding method can be represented by the firing rate v as follows:

$$v=\frac{N_{\text{spike}}}{T}, \tag{1}$$

where N_{spike} represents the number of spikes and T denotes the time step. For remote sensing images, the pixel values correspond to the numbers of spikes. However, the spike sequences obtained based on frequency are not sparse, necessitating the resampling of spike sequences from original $\boldsymbol{X}$ to $\boldsymbol{X}$. If we use the L2 norm to measure the error in the resampling result, we can model the concept of compressed sensing as follows:

$$E_m(\boldsymbol{X})=\inf_{S_m\cdot\boldsymbol{\Phi}_m}\sup_{\boldsymbol{x}\in\boldsymbol{X}}\left\|\boldsymbol{x}-R_m\left(\boldsymbol{\Phi}_m(\boldsymbol{x})\right)\right\|_2, \tag{2}$$

where $\boldsymbol{\Phi}_m$ represents a compressed sampling operator in the sparse domain R_m, m denotes the number of non-adaptive sampling samples, and S_m represents the set of all possible m-dimensional sparse vectors. The sampling process for our sparse signal $\boldsymbol{X}$ can be expressed as

$$r(i)=\left\langle \boldsymbol{x},\varphi_i\right\rangle+\varepsilon_i, i=1,2,\cdots,n, \tag{3}$$

where φ_i represents the i^{th} known reference sampling kernel function, n denotes the number of samples, and ε_i indicates the sampling error for the i^{th} sample.

Because of the complex and redundant features of remote sensing images, to minimize information loss during compression, we need to identify a K-order restricted isometry constant (RIC) δ_K that satisfies Eq. (4), ensuring that it is as small as possible to guarantee the robustness of information sampling:

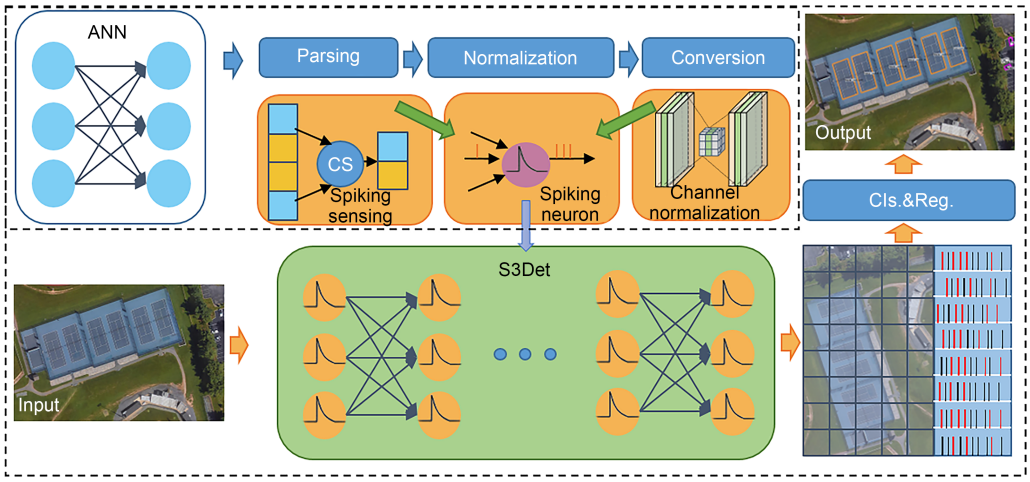

Fig. 2 Architecture of the proposed S3Det detector. In the input image, the object to be detected is a tennis court. ANN: artificial neural network; CS: compressed sensing; Cls.&Reg.: classification and regression

$$\left(1-\delta_K\right)\|\boldsymbol{x}\|_2^2\leq\|\boldsymbol{\Phi x}\|_2^2\leq\left(1+\delta_K\right)\|\boldsymbol{x}\|_2^2 \text{ s.t. } \delta_K<1. \quad (4)$$

However, computing the K-order RIC constant of a matrix is an NP-hard problem. To address this, we draw inspiration from neuroscience, where the visual cortex of the brain can encode visual information with a minimal number of neurons. Thus, we consider using a group sparsity model that incorporates prior knowledge of the image to constrain the problem. The sparsity model can be defined as follows:

$$\boldsymbol{X}_{S,K}^{g}=\left\{\boldsymbol{x}:\sum_{i=1}^{|\boldsymbol{G}|}\delta\left(\left\|\boldsymbol{x}_{n_i}\right\|_q\right)\leq K, g=\left\{\Omega_i\right\}_{i=1}^{|\boldsymbol{G}|}, K\ll|\boldsymbol{G}|\right\}, \quad (5)$$

where g denotes a group structure, in which the components within the structure are correlated, S denotes the set of all possible sparse vectors, $\boldsymbol{G}$ represents a matrix whose column vectors indicate the sparse representation of the signal, q denotes a specific group used to define sparsity constraints, and Ω typically represents an index set used to select specific groups within the signal.

In the initial spike sequence, there is a strong correlation between the spike count and pixel intensity. As inherent attributes of images, non-local similarity and group sparsity are widely used in image reconstruction. Therefore, we use a Gaussian mixture model (Komárek and Lesaffre, 2008) to capture the prior knowledge of the image. First, for a given remote sensing image patch, we identify the $p-1$ most similar image patches to form the image patch group $\boldsymbol{X}_p$. Subtracting the group mean from these patches yields the corresponding residual group $\boldsymbol{X}_n$. Assuming that the residuals of the same matrix belong to the same Gaussian component and are independent between matrices, the likelihood function of $\boldsymbol{X}_n$ can be expressed as follows:

$$\Pr\left(\bar{\boldsymbol{X}}_n\right)=\sum_{k=1}^{K}\pi_k\prod_{m=1}^{M}\mathcal{N}\left(\bar{\boldsymbol{x}}_{m,n}\middle|\mu_k,\Sigma_k\right), \quad (6)$$

where $\mathcal{N}\left(\bar{\boldsymbol{x}}_{m,n}\middle|\mu_k,\Sigma_k\right)$ is the k^{th} component of the Gaussian mixture, and π_k is the normalized mixture coefficient.

After initializing the coefficients in Eq. (6), we use expectation–maximization (E-M) for solving the equation. By alternately executing these two steps until the function converges, we can learn K Gaussian components. Their covariance matrices represent feature information. The image processing procedure is shown in Fig. 3.

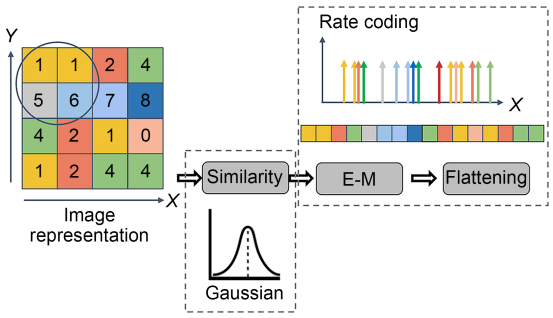

Fig. 3 Processing diagram of the image Gaussian distribution (E-M: expectation–maximization)

Next, we can use the learned prior information to substitute into the group sparse model in Eq. (5) and conditionally relax the sampling operator $\boldsymbol{\Phi}$. We consider the model's block coherence and within-group correlation $v(\boldsymbol{\Phi})$:

$$\mu_B(\boldsymbol{\Phi})=\max\frac{1}{d}\left\|\boldsymbol{\Phi}_{1,\Omega_i}^{\mathrm{T}}\boldsymbol{\Phi}_{1,\Omega_j}\right\|_2\leq\mu(\boldsymbol{\Phi}), \quad (7)$$

$$v(\boldsymbol{\Phi})=\max_i\ \max_{\forall p\neq q}\left|\boldsymbol{\phi}_p^{\mathrm{T}}\boldsymbol{\phi}_q\right|, \quad (8)$$

where μ_B describes the global properties of $\boldsymbol{\Phi}$, including the local similarity and smoothness of the remote sensing image. $\boldsymbol{\phi}_q$ is the q^{th} column vector in $\boldsymbol{\Phi}_{1,\Omega_i}^{\mathrm{T}}$. According to Eldar and Kutyniok (2012), $\boldsymbol{\Phi}$ needs only to satisfy inequality (9) to correctly perceive the original signal $\boldsymbol{X}$:

$$K<\frac{1}{2}\left(\frac{1}{d\mu_B(\boldsymbol{\Phi})}+1+\left(\frac{1}{d}-1\right)\frac{v(\boldsymbol{\Phi})}{\mu_B(\boldsymbol{\Phi})}\right). \quad (9)$$

When $\boldsymbol{X}\in\boldsymbol{X}_{S,K}^{g}$, the perception process can be converted into solving the following mixed norm minimization problem:

$$\begin{cases}P_{q,2}^{S}(\boldsymbol{\Phi},g,\varepsilon):\boldsymbol{x}\leftarrow\arg\min_x\|\boldsymbol{x}\|_{g,q,2} \text{ s.t. } \|\boldsymbol{r}-\boldsymbol{\Phi x}\|_2\leq\varepsilon,\\ P_{q,2}^{S}(\boldsymbol{\Phi},g,K):\boldsymbol{x}\leftarrow\arg\min_x\|\boldsymbol{r}-\boldsymbol{\Phi}\|_2 \text{ s.t. } \|\boldsymbol{x}\|_{g,q,2}\leq K.\end{cases} \quad (10)$$

When q=1, it is the group sparsity-induced norm. Through iterative shrinkage–thresholding methods, we can minimize the mixed norm, promoting simultaneous zeros within each group to achieve group sparsity.

Thus, we demonstrate that the proposed group sparse model can effectively perform compressive sampling of the original signal and achieve high-probability recovery within compressive sensing. Fig. 4 illustrates the difference between our group sparse sampling sequence and the regular sampling sequence.

3.3 Channel self-decaying normalization based on low-order statistics

Unlike CNNs, which can process arbitrary inputs using a sliding window approach, SNNs require neurons to generate spike sequences based on input magnitude. In this context, weights and threshold voltages are responsible for ensuring the adequacy and balance of neuron activation, respectively. However, during processing, neurons may become underactivated or overactivated, leading to information loss and poor performance. Therefore, in SNNs, we apply appropriate normalization techniques to the input signals. Common normalization methods include batch normalization (BN) and layer normalization (LN). BN, originally designed for deep ANNs, can be adapted for use in SNNs. By normalizing the input features across the entire batch, a BN facilitates faster convergence and may improve generalizability. LN, on the other hand, normalizes the inputs across all units within a single sample, making it particularly useful for handling variable-length sequence data. However, when addressing normalization errors, spiking-YOLO identifies that the layer norm significantly reduces neuron firing rates, causing underactivation. Therefore, spiking-YOLO employs fine-grained channel normalization to ensure that even minimal activation values are properly normalized. The channel normalization formula in spiking-YOLO is as follows:

$$\tilde{w}^l_{c_{\text{in}},c_{\text{out}}}=w^l_{i,j}\frac{\lambda^{l-1}_i}{\lambda^l_j},\quad \tilde{b}^l_j=\frac{b^l_j}{\lambda^l_j}, \tag{11}$$

where w^l, λ^l, and b^l denote the weight, the maximum activation calculated from the training dataset, and the bias in convolutional layer l, respectively. c represents the number of channels.

However, further analysis reveals that while channel normalization addresses the underactivation issue, it does not resolve—and even exacerbates—the overactivation problem. This is caused by the SIN error. In fact, the SIN error already exists with the more commonly used layer norm in image classification. Since current conversion uses real values for input, assuming no neuron activation values exceed 1, the firing rate output by the encoding layer, which receives a constant current over sufficient time steps, can exactly match the ANN's activation value. This can be expressed as

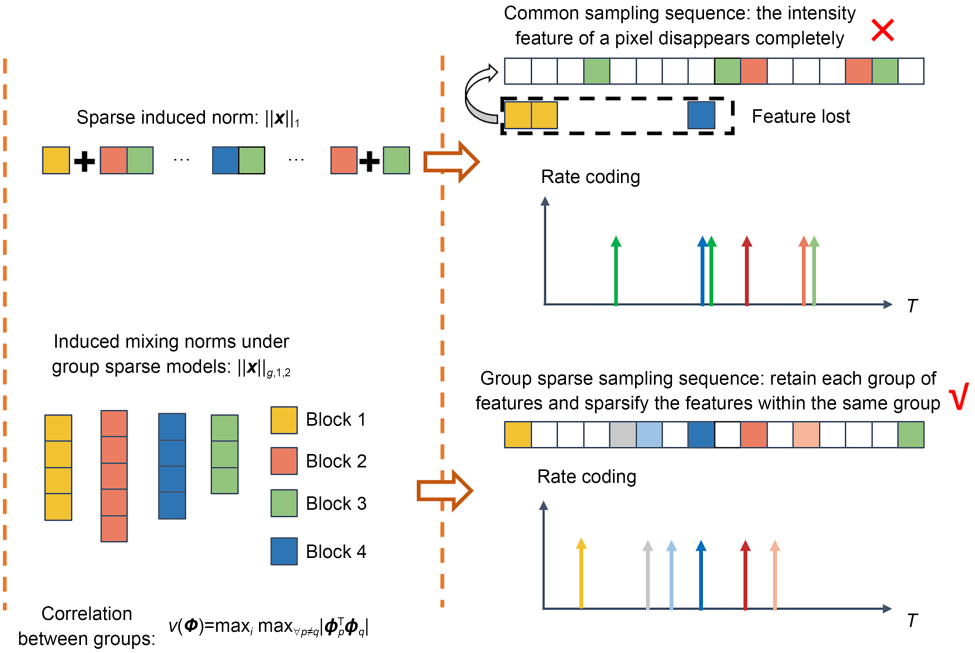

Fig. 4 Diagram of the group sparse sampling sequence and regular sampling sequence

$$r_i^{(l)}(T)=\left\lfloor \frac{\sum_{t=0}^{T}\sum_{j=0}^{J} S_{i,j}^{(l)}(t)}{T} \right\rfloor, \tag{12}$$

where $S_{i,j}^{(l)}$ is the synaptic current of a single layer l and $r_i^{(l)}(T)$ is the activation function value.

After the first layer, the neurons receive discrete spikes. According to Li et al. (2022), disregarding the error from the floor function, the number of emitted spike is the time-varying maximum of the synaptic current sum for each layer's neurons. Thus, the total number of spikes $S_i^{(l)}(T)$ over time step T equals $\max_{0\le t\le T}\left\{\left\lfloor \sum_{\tau=0}^{t} I_i^{(l)}(\tau) \right\rfloor\right\}$, where I is the synaptic current from layer 0 to layer l. When the activation function value $o_i^{(k)}$ of the i^{th} neuron in the k^{th} layer is less than 0 but the synaptic current exceeds the threshold, the neuron will still fire spikes even though it should not be activated in the ANN. This results in the SIN error. The spike error in the k^{th} layer can be expressed as

$$\Delta E_i^{(k)}=\frac{S_i^{(k)}(T)}{T}-o_i^{(k)}\approx\frac{1}{T}\left\lfloor \sum_{j\in \mathrm{SIN}} W_{ij}^{(k)} \right\rfloor. \tag{13}$$

Since the firing rate in SNNs provides high variance while the activation function values in ANNs provide a high mean, the mathematical distributions of their hidden layers differ. Additionally, similar to error accumulation in ANNs, the SIN problem becomes increasingly pronounced with more layers and channels. This issue is particularly significant for neural networks dealing with remote sensing images and cannot be ignored. For example, R3Det has 4096 channels, and the accumulation of errors across these numerous channels significantly increases the conversion error in the resulting SNN. To address this issue, spike calibration (Li et al., 2022) was proposed, using a spike monitor to calculate the interspike interval (ISI) and neurons with negative weights to counteract the activation of inactive neurons. However, this method unnecessarily increases the number of neurons, leading to additional computational overhead.

To address the SIN error in a cost-effective manner, we propose an exponential momentum decay scheme based on low-order statistics, building on existing channel normalization in spiking-YOLO, as shown in Fig. 5. This method limits excessive neuron firing rates. We denote the mean and variance of neuron activation in the k^{th} layer and c^{th} channel as $\bar{\mu}_{k,c}$ and $\{\bar{\sigma}^2\}_{k,c}$, respectively. Introducing a momentum decay factor $\eta\in(0,1)$, the mean and variance are progressively updated as the layer depth increases:

$$\bar{\mu}_{k+1,c}=(1-\eta^k)\mu_{k,c}+\eta^k\bar{\mu}_{k,c}, \tag{14}$$

$$\{\bar{\sigma}^2\}_{k+1,c}=(1-\eta^k)\{\sigma^2\}_{k,c}+\eta^k\{\bar{\sigma}^2\}_{k,c}, \tag{15}$$

where $\eta^k=\eta^0\exp(-k)$ is the object momentum coefficient with adaptive decay.

In light of this, as the network depth increases, the normalized weights gradually decrease. This reduction is not abrupt but occurs in a smooth manner, making the process more biologically plausible. Our method effectively suppresses abnormally high firing rates of neurons, leading to more faithful information processing and reduced unnecessary energy consumption. Importantly, we achieve this by adjusting only the weight decay parameters. This approach eliminates overactivated neurons without significantly increasing the computational burden. Consequently, our technique is both efficient and practical, making it well-suited for large-scale network deployments in real-world applications.

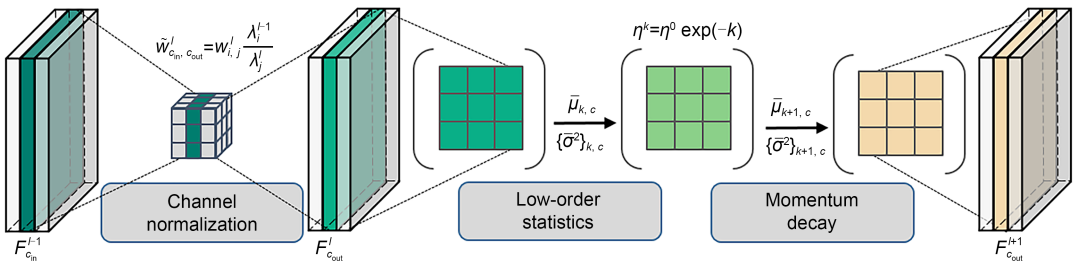

Fig. 5 Diagram of the exponential momentum attenuation scheme. *F* denotes the feature map

4 Experiments

We evaluated the performance of the proposed method for object detection, using R3Det as the initial algorithm before conversion. The subsequent SNN conversion was performed by the SNNToolbox. In addition to standard model parsing, normalization, and simulation procedures, we implemented the rapid perception model and channel self-decay normalization on the Toolbox platform. Max pooling and BN were implemented according to the method proposed by Rueckauer et al. (2017). We conducted our experimental analysis using two publicly available datasets annotated with OBBs: DOTA (Xia et al., 2018) and HRSC2016 (Liu ZK et al., 2016). The experimental setup is detailed in Table 1. In inference, we used rate coding to encode the input images to the input spikes by the integrate-and-fire (IF) neurons.

Table 1 Experimental environment

Category	Description
Server	Inspur NF5466M5
CPU	Intel® Xeon®
GPU	NVIDIA Tesla V100
Memory	253 GiB
OS	Debian 11 Bullseye
SNN framework	SNNToolbox (Rueckauer and Liu, 2021)
Language	Python3.7

4.1 Datasets and settings

DOTA is a large open remote sensing image benchmark dataset comprising thousands of images from different platforms and sensors such as Google Earth and the GF-2 and JL-1 satellites. DOTA-v1.0 contains 2806 images ranging in size from 800×800 pixels to 4000×4000 pixels, and the dataset is divided into a training set, a validation set, and a test set at a ratio of 3:1:2. The images cover 15 object types with a total of 188 282 objects, including helicopter (HC), swimming pool (SP), harbor (HA), roundabout (RA), soccer-ball field (SBF), storage tank (ST), basketball court (BC), tennis court (TC), ship (SH), large vehicle (LV), small vehicle (SV), ground track field (GTF), bridge (BR), baseball diamond (BD), and plane (PL). We trained the model for 36 rounds (183 600 iterations).

The high-resolution ship collection 2016 (HRSC2016) dataset is used mainly for various ship detection tasks. It labels three major categories of ships: aircraft carriers, warcrafts, and merchant ships. Within the three categories, there are 27 subcategories of objects, with a total of 2976 objects. The training set includes 436 images with 1207 samples, the validation set includes 181 images with 541 samples, and the test set includes 444 images with 1228 samples.

4.2 Efficiency of S3Det

To validate and analyze the efficiency of the proposed method, we examined the impact of the SSRS module and CSWN on both performance and energy consumption.

4.2.1 Detection efficiency

First, we evaluated the detection efficiency, as image processing speed is a crucial metric for real-time applications on embedded devices such as drones. We compared the speed of S3Det with those of two categories of algorithms: common remote sensing image detection algorithms and lightweight detection algorithms suitable for rotated objects. Additionally, to increase detection speed on edge devices, we replaced S3Det's backbone with MobileNetV2 (Sinha and El-Sharkawy, 2019) and ShuffleNetV2 (Ma et al., 2018) and measured the relevant performance indicators.

"#Params" represents the total number of model parameters. "Ratio" indicates the proportion of the actual runtime consumed by the model in relation to the overall inference time taken to process 1000 sub-images. This ratio was computed using the standardized method, provided by TorchStat. Meanwhile, the speed (in frames per second) was measured using MMDetection (Chen K et al., 2019), and the testing environment comprised four Tesla V100 GPUs with a batch size of 16. The detection results of the algorithm are shown in Table 2.

As shown in Table 2, using ResNet as the backbone, S3Det with a stride of 64 significantly improved detection speed compared to the ANN model (R3Det). Specifically, using ResNet50 as the backbone network, the detection speed of S3Det was 20 frames/s, representing a 42.86% increase over that of R3Det. Additionally, by using a more lightweight backbone, such as ShuffleNet, the detection speed can be rapidly

Table 2 Speed comparison on DOTA and HRSC2016

Model	Backbone	Image size	DOTA			HRSC2016		
			#Params (MiB)	Ratio (%)	Speed (frame/s)	#Params (MiB)	Ratio (%)	Speed (frame/s)
R3Det (Yang et al., 2021a)	ResNet50	800×800	485	88.52	14	496	89.42	8
SCRDet (Yang et al., 2019)	ResNet50	800×800	452	71.20	10	484	76.03	4.5
R^2CNN (Jiang YY et al., 2018)	ResNet50	600×600	353	93.60	–	314	94.41	2
RRPN (Ma et al., 2018)	ResNet50	600×600	348	94.30	5	306	92.43	3.5
RetinaNet-R (Lin et al., 2017)	ResNet50	800×800	378	82.85	12	366	83.90	10
MobileDets (Xiong et al., 2021)	MobileNetV3	300×300	96	31.78	41.5	104	35.98	33
GiraffeDet-R (Jiang YQ et al., 2022)	S2D Chain	300×300	137	35.32	35	153	53.64	24
OR-CNN (Xie et al., 2024c)	ResNet50	800×800	368	37.31	15.3	269	78.43	14.9
DFDet (Xie et al., 2024b)	ResNet50	800×800	392	36.82	23.4	280	84.52	21
YOLOv10-L (Wang et al., 2024)	CSPNet	300×300	152	40.73	32	96	31.22	32
S3Det (step=64)	ResNet50	800×800	462	64.47	20	482	66.59	14
	ResNet101	800×800	758	72.57	12	684	77.22	8
	ResNet152	800×800	961	85.29	8	887	89.17	6
	MobileNetV2	300×300	168	32.14	35	121	33.98	31
	ShuffleNetV2	300×300	147	30.47	41	107	32.10	34

The speed of R^2CNN is <1 frame/s, so it is indicated by "–"

increased to 41 frames/s, further demonstrating S3Det's efficiency. We also note that the inference time ratio of the S3Det model represented a significant reduction over that of the baseline model (DOTA: 64.47% versus 88.52%; HRSC2016: 66.59% versus 89.42%). However, the model's parameter count did not substantially decrease. This may be attributed to the SNN model parameters being largely inherited from the ANN, with our SSRS primarily reducing input redundancy, resulting in no substantial reduction in the parameter count.

4.2.2 Energy efficiency

SNNs exhibit exceptionally low power consumption due to their event-driven neural activity and rich spatio-temporal dynamics. In our approach, the sparsity gain introduced by the SSRS further enhances the optimization of the model's operational power consumption. To thoroughly understand the role of sparsity in SNN, we evaluated the power consumption metric of the R3Det model before conversion and the S3Det model after conversion. Before conversion, we divided R3Det into four subnetworks: ResNet, FPN, RPN, and RefineNet. Fig. 6 presents the number of multiply-added nodes for each subnetwork. The total multiply-accumulate (MAC) operations for all networks amount to 2.15×10^{11}.

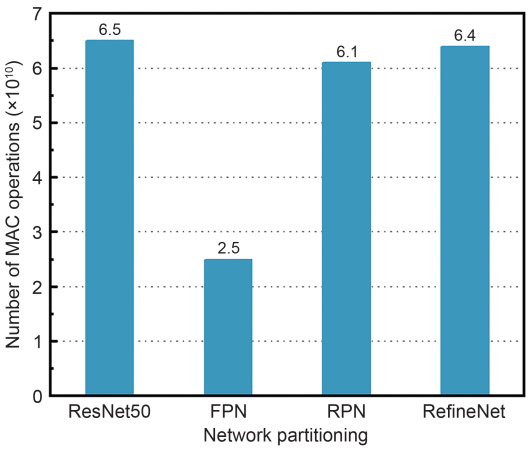

Fig. 6 Number of MAC operations of the subnetwork

For the post-conversion, we defined one time step as 1 ms (1 kHz synchronization signal in Merolla et al. (2014)). According to Horowitz (2014), the energy cost per operation was 4.6 pJ for FLOAT32 MAC and 0.9 pJ for FLOAT32 AC. Considering that S3Det accepts analog image inputs, we defined the first layer as MAC operations. Rathi and Roy (2023) argued that the ratio of ANN to SNN energy consumption can be expressed as

$$\frac{E_{\mathrm{ANN}}}{E_{\mathrm{SNN}}}=\frac{\mathrm{OP}_{\mathrm{ANN}}\cdot E_{\mathrm{MAC}}}{\mathrm{OP}_{\mathrm{SNN}}\cdot E_{\mathrm{AC}}}=\frac{\mathrm{OP}_{\mathrm{ANN}}\cdot E_{\mathrm{MAC}}}{\mathrm{OP}_{\mathrm{ANN}}\cdot\text{spike rate}\cdot E_{\mathrm{AC}}}, \quad (16)$$

where E_{ANN}, E_{SNN}, and E_{MAC} denote the energy consumption of ANN, SNN, and a single MAC operation, respectively. OP_{ANN} and OP_{SNN} represent the total number of operations in ANN and SNN, respectively. E_{AC} denotes the energy consumption of a floating-point plus.

Since S3Det was converted from the ANN network R3Det, we calculated the energy consumption of S3Det using the following formula:

$$E_{S3Det}=OP_{S3Det}\cdot\text{spike rate}\cdot E_{AC}, \tag{17}$$

where E_{S3Det} and OP_{S3Det} denote the energy consumption and the number of operations of S3Det, respectively.

We recorded the number of SNN operations and the average spike rate during S3Det's inference over a specific time step. To facilitate comparison, we included R3Det's power consumption metric and calculated the detailed energy consumption, as summarized in Table 3.

Our calculations in Table 3 indicate that when the input was a 32-bit floating point, using CSWN enabled the S3Det model to reduce energy consumption to approximately 1/21 and power consumption to approximately 1/122 of R3Det's. This demonstrates the significant low-power advantage of S3Det. Additionally, our proposed channel self-decaying normalization method reduced the spike rate by 15.44%, effectively addressing the issue of abnormal neuron activation and limiting excessive firing rates. This reduction in spike rate further contributed to the overall decrease in the model's power consumption.

4.3 High-precision detection experiments

Recent studies (Li et al., 2022; Hu et al., 2023) have shown that the performance of converted SNNs can match or even exceed that of ANNs in natural image object detection. However, these algorithms have not performed as well on remote sensing data. In our accuracy experiments, we used average precision (AP) and mean average precision (mAP), which are standard metrics in object detection. In this subsection, we present the results of S3Det on DOTA and HRSC2016. As a benchmark, we conducted comparative experiments of R3Det and S3Det using ResNet-50, ResNet-101, and ResNet-152. In addition, we evaluated the detection accuracy of several commonly used algorithms.

On DOTA, our proposed spiking conversion method is based on the R3Det benchmark. Hence, we selected comparison methods that have frequently been used with R3Det in recent studies, including high-precision algorithms such as one-stage detectors DAL (Ming et al., 2021) and S^2A-Net (Han et al., 2022) and two-stage detectors ICN (Azimi et al., 2018), CAD-Net (Zhang GJ et al., 2019), OR-CNN (Xie et al., 2024c), and DFDet (Xie et al., 2024b). On HRSC2016, RC1 (Liu ZK et al., 2016) and RRD (Yang and Yan, 2022) used VGG16 as the backbone, while the remaining algorithms used ResNet101. Different methods use various input image sizes. To highlight the efficiency advantages, we did not use the most accurate backbone networks for the comparison, as this would significantly increase the detection time with only marginal gains in accuracy. Instead, we opted for typical configurations to provide a fair and practical comparison. The experimental results are shown in Tables 4 and 5.

The accuracy experiments revealed that S3Det achieved strong detection performance on both the DOTA and HRSC2016 datasets. With a stride of 512 and ResNet-152 as the backbone, S3Det attained 70.33% mAP on DOTA, which was only a reduction of 3.31 percentage points compared to the original algorithm (73.64%). While the theoretical conversion from ANN to SNN was designed to be lossless, a significant mismatch between the firing rates and the activation values

Table 3 Energy consumption comparison on DOTA

Method	Data type	Input	FLOPs	OP_{S3Det}	Spike rate	Energy (J)	Power (W)
R3Det	32-bit float	800×800	4.334E+11	–	–	1.994	178
S3Det (Channel norm)*	32-bit float	800×800	–	4.275E+11	39.76%	1.53E−01	2.39
S3Det (Channel self-decaying norm)*	32-bit float	800×800	–	4.275E+11	24.32%	9.36E−02	1.46

In SNNs, model complexity is measured using spike operations and firing rates. Therefore, the floating point operations per second (FLOPs) metric is not applicable and is left as "–" in the table. In contrast, for ANNs, FLOPs remains the standard metric for measuring computational complexity.
* time step=64

Table 4 Evaluation of the OBB task on the DOTA testing set

Method		AP (%)								mAP (%)
		PL	BD	BR	GTF	SV	LV	SH	TC	
One-stage	RetinaNet-R	88.92	67.67	33.55	56.83	66.11	73.28	75.24	90.87	62.02
	DAL	88.68	76.55	45.08	66.80	67.00	76.76	79.74	90.84	71.45
	S^2A-Net	89.11	82.84	48.37	71.11	78.11	78.39	87.25	90.83	74.12
	R3Det-50	89.30	80.29	46.21	65.07	70.51	73.38	77.42	90.83	70.16
	R3Det-101	89.54	81.99	48.46	62.52	70.48	74.29	77.54	90.80	71.69
	R3Det-152	89.42	81.03	50.41	65.93	70.90	78.63	78.03	90.67	73.64
Two-stage	SCRDet	89.98	80.65	52.09	68.36	64.52	60.32	72.41	90.85	72.36
	R^2CNN	80.94	65.67	35.34	67.44	59.92	50.91	55.81	90.67	60.67
	RRPN	88.52	71.20	31.66	59.30	51.85	56.19	57.25	90.81	61.21
	ICN	81.36	74.30	47.70	70.32	64.89	67.82	69.98	90.76	68.16
	CAD-Net	87.80	82.40	49.40	73.50	71.10	63.50	76.60	90.90	69.88
	OR-CNN	89.46	82.12	54.78	70.86	78.93	83.00	88.20	90.90	75.88
	DFDet	88.92	79.25	48.40	70.00	80.22	78.85	87.21	90.90	74.71
Ours*	S3Det-50	87.83	77.45	34.05	64.50	62.34	73.10	66.11	90.56	66.72
	S3Det-101	89.16	77.79	43.62	58.11	66.56	70.99	72.22	86.89	67.74
	S3Det-152	88.92	80.10	48.01	62.65	70.31	71.48	72.23	85.95	70.33

Method		AP (%)							mAP (%)
		BC	ST	SBF	RA	HA	SP	HC	
One-stage	RetinaNet-R	73.95	75.07	43.77	56.72	51.05	55.86	21.46	62.02
	DAL	79.54	78.45	57.71	62.27	69.05	73.14	60.11	71.45
	S^2A-Net	84.90	85.64	60.36	62.60	65.26	69.13	57.94	74.12
	R3Det-50	80.59	82.26	59.29	58.25	57.75	65.90	55.31	70.16
	R3Det-101	81.39	83.54	61.97	59.82	65.44	67.46	60.05	71.69
	R3Det-152	85.24	84.10	61.64	63.52	68.15	69.80	67.09	73.64
Two-stage	SCRDet	87.94	86.86	65.02	66.68	66.25	68.24	65.21	72.36
	R^2CNN	66.92	72.39	55.06	52.23	55.14	53.35	48.22	60.67
	RRPN	72.84	67.38	59.69	52.84	53.08	51.94	53.58	61.21
	ICN	79.06	78.20	53.64	62.90	67.02	64.17	50.23	68.16
	CAD-Net	79.20	73.30	48.40	60.90	62.00	67.00	62.20	69.88
	OR-CNN	87.50	84.68	63.97	67.69	74.94	68.84	52.28	75.88
	DFDet	83.13	83.98	60.07	66.49	68.27	76.78	58.22	74.71
Ours*	S3Det-50	75.33	78.28	55.37	56.12	62.88	64.11	52.84	66.72
	S3Det-101	77.34	79.36	57.41	55.11	60.90	63.78	56.89	67.74
	S3Det-152	80.77	81.89	60.44	56.00	65.36	67.80	63.11	70.33

PL: plane; BD: baseball diamond; BR: bridge; GTF: ground track field; SV: small vehicle; LV: large vehicle; SH: ship; TC: tennis court; BC: basketball court; ST: storage tank; SBF: soccer-ball field; RA: roundabout; HA: harbor; SP: swimming pool; HC: helicopter. AP: average precision; mAP: mean average precision. * time step=512

persisted. This discrepancy often led to the converted SNNs achieving lower detection accuracy than the ANNs did. However, other than scenarios demanding extremely high precision, the detection performance of S3Det was suitable for most tasks. Thus, the accuracy loss was deemed acceptable. The detection results are visualized in Fig. 7. We also plotted the AP change curves for each category at different time steps, and, as shown in Fig. 8, the APs of all categories were positively correlated with the growth of T and eventually converged to a threshold value.

For the HRSC2016 dataset, we conducted a comparative validation of our method's detection performance by varying the time steps. Fig. 9 shows the

Table 5 Accuracy and speed comparison on HRSC2016

Method	Backbone	Image size	mAP (%)	Speed (frame/s)
R^2CNN	ResNet101	600×600	73.07	2.4
RRPN	ResNet101	600×600	79.08	3.7
RC1	VGG16	800×800	75.71	–
RRD	VGG16	384×384	84.32	–
RoI-Transformer (Ding et al., 2019)	ResNet101	512×800	86.24	6.1
R3Det	ResNet101	800×800	89.26	10.6
S3Det*	ResNet101	800×800	85.24	12.4

"–" means <1 frame/s. * time step=512

detection advantage of our method as the number of time steps increased. When T=256, S3Det successfully detected three types of ships, whereas Spiking-R3Det, converted directly from R3Det, succeeded only at 1280 steps. Additionally, at this time step, the bounding box for the merchant ship (pink) was inaccurately drawn. We hypothesize that this is due to the group sparse model, which sparsifies the object pixels, thereby reasonably reducing the processing time occupied by redundant pixels. Additionally, the number of invalid boxes produced by the channel norm was significantly greater than that produced by the channel self-decaying norm. We believe this is because the channel self-decaying norm effectively suppresses the SIN error and mitigates abnormal spiking, thereby reducing the occurrence of erroneous pixel detection.

5 Conclusions

In this work, we develop a comprehensive and efficient conversion framework for detecting remote sensing images while maintaining low power consumption. Our method achieves high precision with fewer time steps compared to benchmark methods, while having significantly lower power consumption than does ANN.

In fact, our conversion method can be applied to almost all ANNs. The experimental results from object detection tasks demonstrate that our conversion method can significantly reduce energy consumption while maintaining or even improving performance. To evaluate versatility across different scenarios, we are currently testing the converted SNNs in various settings, including real-time mobile applications, embedded systems, and edge computing environments. However, the conversion process may not always preserve the exact behavior of the original ANN, especially when the ANN

Fig. 7 Visualization of S3Det (time step=512) on DOTA. References to color refer to the online version of this figure

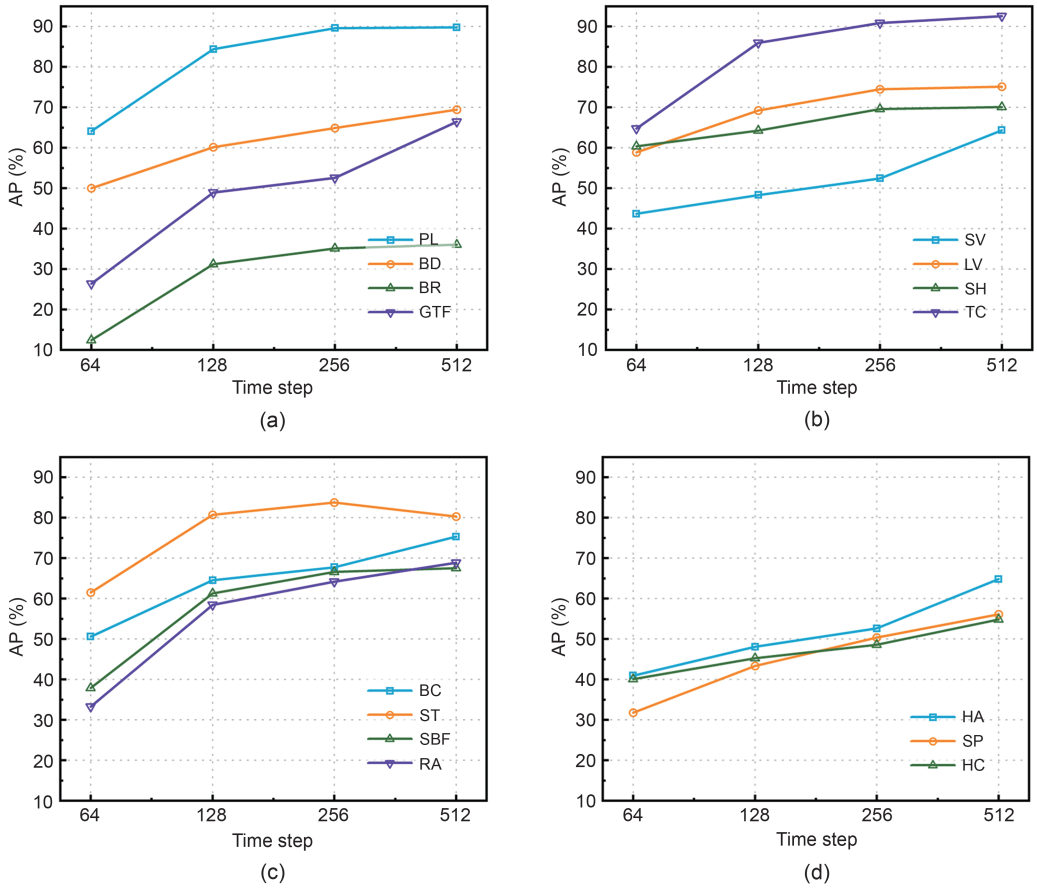

Fig. 8 The AP for each category at different time steps: (a) PL, BD, BR, and GTF; (b) SV, LV, SH, and TC; (c) BC, ST, SBF, and RA; (d) HA, SP, and HC

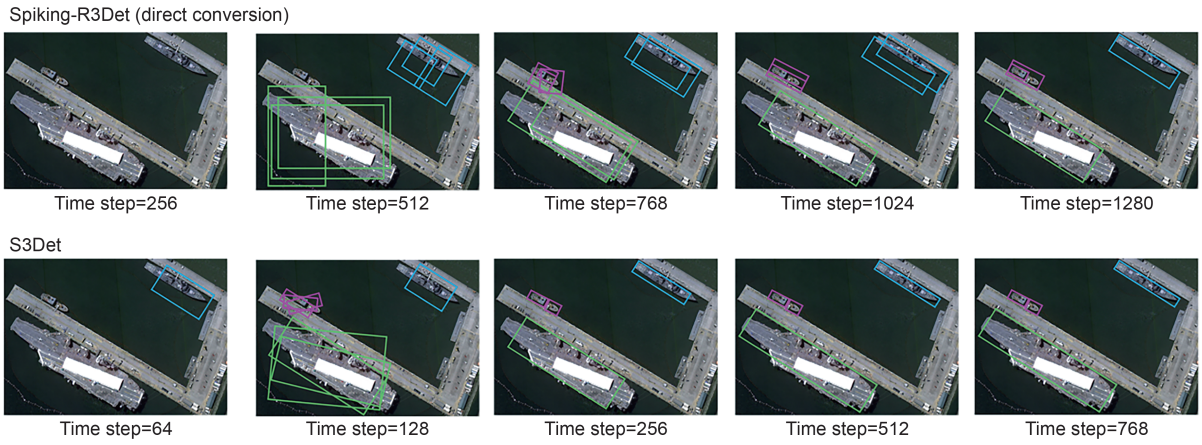

Fig. 9 Spiking-R3Det and S3Det visualization results on HRSC2016. Pink represents a merchant ship, green represents an aircraft carrier, and blue represents a warcraft. References to color refer to the online version of this figure

relies heavily on fine-tuned weights and complex non-linearities. Additionally, the conversion process may introduce a small overhead in terms of computational resources during the initial setup, which could be a consideration for extremely resource-constrained applications. To further enhance the generalizability of our method, future work will focus on direct training and inference of SNNs.

Contributors

Li CHEN led the design of the experiments, participated in the data analysis and interpretation of the results, and drafted the paper. Fan ZHANG constructed the theoretical model and conducted a comprehensive review of the literature. Guangwei XIE executed the experiments and collected the data. Yanzhao GAO organized the theoretical knowledge into mathematical formulations. Xiaofeng QI drafted and revised the paper. Mingqian SUN managed the collection and organization of preliminary research materials.

Conflict of interest

All the authors declare that they have no conflict of interest.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

Azimi SM, Vig E, Bahmanyar R, et al., 2018. Towards multi-class object detection in unconstrained remote sensing imagery. Proc 14th Asian Conf on Computer Vision, p.150-165. https://doi.org/10.1007/978-3-030-20893-6_10

Chen GH, Pei GS, Tang Y, et al., 2022. A novel multi-sample data augmentation method for oriented object detection in remote sensing images. Proc IEEE 24th Int Workshop on Multimedia Signal Processing, p.1-7. https://doi.org/10.1109/MMSP55362.2022.9949615

Chen K, Wang JQ, Pang JM, et al., 2019. MMDetection: open MMlab detection toolbox and benchmark. https://arxiv.org/abs/1906.07155

Chen L, Zhang F, Guo W, et al., 2023. SFTN: fast object detection for aerial images. *IET Image Process*, 17(13):3897-3907. https://doi.org/10.1049/ipr2.12906

Cheng G, Zhou PC, Han JW, et al., 2016a. Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images. *IEEE Trans Geosci Remote Sens*, 54(12):7405-7415. https://doi.org/10.1109/TGRS.2016.2601622

Cheng G, Zhou PC, Han JW, 2016b. RIFD-CNN: rotation-invariant and Fisher discriminative convolutional neural networks for object detection. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.2884-2893. https://doi.org/10.1109/CVPR.2016.315

Ding J, Xue N, Long Y, et al., 2019. Learning RoI Transformer for oriented object detection in aerial images. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.2844-2853. https://doi.org/10.1109/CVPR.2019.00296

Eldar YC, Kutyniok G, 2012. Compressed Sensing: Theory and Applications. Cambridge University Press, Cambridge, UK. https://doi.org/10.1017/CBO9780511794308

Everingham M, van Gool L, Williams CK, et al., 2010. The PASCAL Visual Object Classes (VOC) challenge. *Int J Comput Vis*, 88(2):303-338. https://doi.org/10.1007/s11263-009-0275-4

Gong MG, Li JZ, Zhang YR, et al., 2022. Two-path aggregation attention network with quad-patch data augmentation for few-shot scene classification. *IEEE Trans Geosci Remote Sens*, 60:4511616. https://doi.org/10.1109/TGRS.2022.3197445

Han JM, Ding J, Li J, et al., 2022. Align deep features for oriented object detection. *IEEE Trans Geosci Remote Sens*, 60:5602511. https://doi.org/10.1109/TGRS.2021.3062048

He X, Ma SP, He LY, et al., 2022. High-resolution polar network for object detection in remote sensing images. *IEEE Geosci Remote Sens Lett*, 19:6000605. https://doi.org/10.1109/LGRS.2020.3039240

Horowitz M, 2014. 1.1 Computing's energy problem (and what we can do about it). Proc IEEE Int Solid-State Circuits Conf Digest of Technical Papers, p.10-14. https://doi.org/10.1109/ISSCC.2014.6757323

Hu YF, Zheng Q, Jiang XD, et al., 2023. Fast-SNN: fast spiking neural network by converting quantized ANN. *IEEE Trans Patt Anal Mach Intell*, 45(12):14546-14562. https://doi.org/10.1109/TPAMI.2023.3275769

Huang ZC, Li W, Xia XG, et al., 2022. A general Gaussian heatmap label assignment for arbitrary-oriented object detection. *IEEE Trans Image Process*, 31:1895-1910. https://doi.org/10.1109/TIP.2022.3148874

Jiang YQ, Tan ZY, Wang JY, et al., 2022. GiraffeDet: a heavy-neck paradigm for object detection. https://arxiv.org/abs/2202.04256

Jiang YY, Zhu XY, Wang XB, et al., 2018. R^2CNN: rotational region CNN for arbitrarily-oriented scene text detection. Proc 24th Int Conf on Pattern Recognition, p.3610-3615. https://doi.org/10.1109/ICPR.2018.8545598

Kim S, Park S, Na B, et al., 2020. Spiking-YOLO: spiking neural network for energy-efficient object detection. Proc 34th AAAI Conf on Artificial Intelligence, p.11270-11277. https://doi.org/10.1609/aaai.v34i07.6787

Komárek A, Lesaffre E, 2008. Generalized linear mixed model with a penalized Gaussian mixture as a random effects distribution. *Comput Stat Data Anal*, 52(7):3441-3458. https://doi.org/10.1016/j.csda.2007.10.024

LeCun Y, Bottou L, Bengio Y, et al., 1998. Gradient-based learning applied to document recognition. *Proc IEEE*, 86(11):2278-2324. https://doi.org/10.1109/5.726791

Li Y, He X, Dong YT, et al., 2022. Spike calibration: fast and accurate conversion of spiking neural network for object detection and segmentation. https://arxiv.org/abs/2207.02702

Lin TY, Maire M, Belongie S, et al., 2014. Microsoft COCO: common objects in context. Proc 13th European Conf on Computer Vision, p.740-755. https://doi.org/10.1007/978-3-319-10602-1_48

Lin TY, Goyal P, Girshick R, et al., 2017. Focal loss for dense object detection. Proc IEEE Int Conf on Computer Vision, p.2999-3007. https://doi.org/10.1109/ICCV.2017.324

Liu WX, Luo B, Liu J, et al., 2022. Synthetic data augmentation using multiscale attention CycleGAN for aircraft detection in remote sensing images. *IEEE Geosci Remote Sens Lett*, 19:4009205. https://doi.org/10.1109/LGRS.2021.3052017

Liu ZK, Wang HZ, Weng LB, et al., 2016. Ship rotated bounding box space for ship extraction from high-resolution optical satellite images with complex backgrounds. *IEEE Geosci Remote Sens Lett*, 13(8):1074-1078.

https://doi.org/10.1109/LGRS.2016.2565705

Ma JQ, Shao WY, Ye H, et al., 2018. Arbitrary-oriented scene text detection via rotation proposals. *IEEE Trans Multim*, 20(11):3111-3122. https://doi.org/10.1109/TMM.2018.2818020

Maass W, 1997. Networks of spiking neurons: the third generation of neural network models. *Neur Netw*, 10(9):1659-1671. https://doi.org/10.1016/S0893-6080(97)00011-7

Merolla PA, Arthur JV, Alvarez-Icaza R, et al., 2014. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197):668-673. https://doi.org/10.1126/science.1254642

Ming Q, Zhou ZQ, Miao LJ, et al., 2021. Dynamic anchor learning for arbitrary-oriented object detection. Proc 35th AAAI Conf on Artificial Intelligence, Electronic Network, p.2355-2363. https://doi.org/10.1609/aaai.v35i3.16336

Rathi N, Roy K, 2023. DIET-SNN: a low-latency spiking neural network with direct input encoding and leakage and threshold optimization. *IEEE Trans Neur Netw Learn Syst*, 34(6):3174-3182. https://doi.org/10.1109/TNNLS.2021.3111897

Rueckauer B, Liu SC, 2021. Temporal pattern coding in deep spiking neural networks. Proc Int Joint Conf on Neural Networks, p.1-8. https://doi.org/10.1109/IJCNN52387.2021.9533837

Rueckauer B, Lungu IA, Hu YH, et al., 2017. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Front Neurosci*, 11:682. https://doi.org/10.3389/fnins.2017.00682

Sinha D, El-Sharkawy M, 2019. Thin MobileNet: an enhanced MobileNet architecture. Proc IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conf, p.280-285. https://doi.org/10.1109/UEMCON47517.2019.8993089

Vaswani A, Shazeer N, Parmar N, et al., 2017. Attention is all you need. Proc 31st Int Conf on Neural Information Processing Systems, p.6000-6010.

Wang A, Chen H, Liu LH, et al., 2024. YOLOv10: real-time end-to-end object detection. https://arxiv.org/abs/2405.14458

Xia GS, Bai X, Ding J, et al., 2018. DOTA: a large-scale dataset for object detection in aerial images. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.3974-3983. https://doi.org/10.1109/CVPR.2018.00418

Xie XX, Lang CB, Miao SC, et al., 2023. Mutual-assistance learning for object detection. *IEEE Trans Patt Anal Mach Intell*, 45(12):15171-15184. https://doi.org/10.1109/TPAMI.2023.3319634

Xie XX, Cheng G, Li QY, et al., 2024a. Fewer is more: efficient object detection in large aerial images. *Sci China Inform Sci*, 67(1):112106. https://doi.org/10.1007/s11432-022-3718-5

Xie XX, Cheng G, Rao CF, et al., 2024b. Oriented object detection via contextual dependence mining and penalty-incentive allocation. *IEEE Trans Geosci Remote Sens*, 62: 5618010. https://doi.org/10.1109/TGRS.2024.3385985

Xie XX, Cheng G, Wang JB, et al., 2024c. Oriented R-CNN and beyond. *Int J Comput Vis*, 132(7):2420-2442. https://doi.org/10.1007/s11263-024-01989-w

Xiong YY, Liu HX, Gupta S, et al., 2021. MobileDets: searching for object detection architectures for mobile accelerators. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.3824-3833. https://doi.org/10.1109/CVPR46437.2021.00382

Yang X, Yan JC, 2022. On the arbitrary-oriented object detection: classification based approaches revisited. *Int J Comput Vis*, 130(5):1340-1365. https://doi.org/10.1007/s11263-022-01593-w

Yang X, Yang JR, Yan JC, et al., 2019. SCRDet: towards more robust detection for small, cluttered and rotated objects. Proc IEEE/CVF Int Conf on Computer Vision, p.8231-8240. https://doi.org/10.1109/ICCV.2019.00832

Yang X, Yan JC, Feng ZM, et al., 2021a. R3Det: refined single-stage detector with feature refinement for rotating object. Proc 35th AAAI Conf on Artificial Intelligence, p.3163-3171. https://doi.org/10.1609/aaai.v35i4.16426

Yang X, Yan JC, Ming Q, et al., 2021b. Rethinking rotated object detection with Gaussian Wasserstein distance loss. Proc 38th Int Conf on Machine Learning, p.11830-11841.

Yao M, Zhao GS, Zhang HY, et al., 2023. Attention spiking neural networks. *IEEE Trans Patt Anal Mach Intell*, 45(8):9393-9410. https://doi.org/10.1109/TPAMI.2023.3241201

Zhang C, Lam KM, Wang Q, 2023. CoF-Net: a progressive coarse-to-fine framework for object detection in remote-sensing imagery. *IEEE Trans Geosci Remote Sens*, 61: 5600617. https://doi.org/10.1109/TGRS.2022.3233881

Zhang GJ, Lu SJ, Zhang W, 2019. CAD-Net: a context-aware detection network for objects in remote sensing imagery. *IEEE Trans Geosci Remote Sens*, 57(12):10015-10024. https://doi.org/10.1109/TGRS.2019.2930982

List of supplementary materials

Functional equivalence of ANN-to-SNN