



Event-driven spiking neural network based on membrane potential modulation for remote sensing image classification

Li-Ye Niu ^a, Ying Wei ^{a,b,*}, Yue Liu ^a

^a College of Information Science and Engineering, Northeastern University, Shenyang 110819, China

^b Information Technology R&D Innovation Center of Peking University, Shaoxing, China



ARTICLE INFO

Keywords:

Spiking neural network
Remote sensing images
Transfer learning
Membrane potential modulation
Spike firing rate

ABSTRACT

Spiking neural network (SNN) based on sparse triggering and event-driven is a hardware-friendly model. SNN can provide an ultra-low power alternative for the deep neural network (DNN) to process remote sensing images. Brain information processing depends on the action potential of neurons. Therefore, the biological rationality of the artificial neural network (ANN) has been questioned. SNN is a more suitable model for brain information processing mechanisms. At present, the SNN obtained by ANN conversion has achieved the best performance in the current image processing tasks. However, the method based on ANN to SNN will have performance loss in the conversion process. Herein, we proposed a spiking neuron threshold-following reset (TF-reset) method and a membrane potential modulation method to reduce the loss of network conversion. We theoretically analyzed the proposed TF-reset and deduced the relationship between spike firing rate and neuron activation. In the experiment, we used an improved VGG-15 architecture combined with the method of transfer learning to apply the model to the classification task of remote sensing images. SNN-VGG-15 based on TF-reset and membrane potential modulation algorithm achieved a classification accuracy of 99.14%, 94.54%, and 95.00% on UCM, RSSCN7, and AID. Our algorithm can not only realize the lossless conversion of SNN but also outperforms the original network in classification performance on UCM and RSSCN7. In addition, our model also has advantages in energy consumption and noise robustness. The algorithm in this paper can provide a reference for the research remote sensing images procession using SNN.

1. Introduction

As one of the representative algorithms of deep learning, artificial neural network (ANN) shows an excellent performance in intelligent tasks such as computer vision, image processing, medical aided diagnosis and natural language processing (Tan et al., 2020b). Deep neural network (DNN) shows great potential in the field of remote sensing image research. As the resolution of remote sensing images increases, the structure of DNN becomes more and more complex. Therefore, DNN needs a powerful hardware computing platform (such as a high-performance GPU) to achieve its excellent performance in the classification tasks of remote sensing image (Esser et al., 2015). As a result, the application of DNN will be greatly limited due to the limited computing resources and energy consumption (Neftci et al., 2013).

ANN processes single, static and continuous activation values, while brain neurons rely on action potentials to process external information (Lee et al., 2016). There are great differences in the information processing mechanism between ANN and brain, and the biological rationality of ANN has also been questioned (Zhou et al., 2019). Inspired by the brain, the spiking neural network (SNN) based on sparse

trigger and event-driven is proposed (Bing et al., 2018). As the third-generation neural network with biological interpretability (Taherkhani et al., 2020), SNN has been widely concerned since it was proposed. The biggest difference between ANN and SNN is that the concept of time is added to the transmission (Kasabov, 2014) and processing of internal information in SNN (Pfeiffer and Pfeil, 2018). The spatio-temporal information in SNN is represented by discrete spike train Wu et al. (2019), which relies on spiking neurons to complete the processing and transmission of information. SNN shows great performance in information processing speed and energy consumption.

Compared with DNN, the SNN-based remote sensing image scene classification method is more reasonable in biology. SNN can realize fast and efficient classification of remote sensing images. The biggest advantage of SNN is that it can make full use of the information based on spatio-temporal events to realize the characteristics of low power consumption and hardware friendliness of the model. When SNN is deployed on the neuromorphological chip, it will provide an ultra-low power alternative for DNN processing remote sensing images. Since the complex dynamics of spiking neurons and their discrete spikes do not

* Corresponding author at: Information Technology R&D Innovation Center of Peking University, Shaoxing, China.
E-mail address: weiyi@ise.neu.edu.cn (Y. Wei).

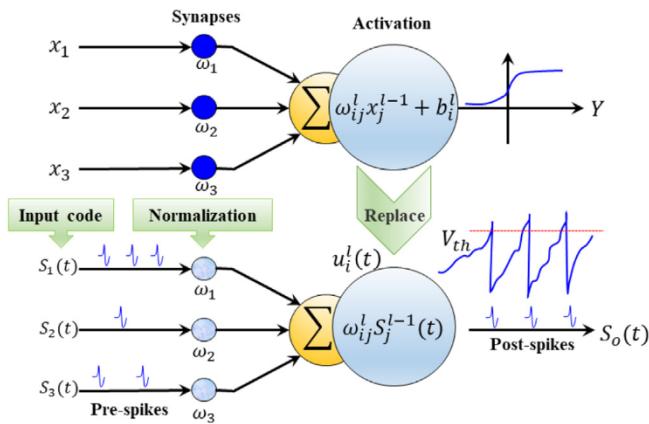


Fig. 1. Schematic diagram of ANN to SNN conversion. (The transformation of the model includes input coding, weight normalization, neuron replacement, and threshold allocation.)

meet the requirements of continuous differentiability, the ANN based on gradient error backpropagation (Tavanaei and Maida, 2019) cannot be directly applied to SNN. It can be seen that the training of SNN cannot be unified, and there are still great difficulties in the direct training of its model. To solve the problem of SNN training, researchers have presented many solutions, which are mainly divided into three categories, shown as follows:

- The first is to train SNN based on spike-time dependent plasticity (STDP). STDP is a local, unsupervised weight correction method with biological interpretability (Kheradpisheh et al., 2018; Mozafari et al., 2019). This method has a great demand for computing resources. Therefore, such methods are currently limited to shallow networks (3–5 layers) to deal with simple classification tasks, and their performance in many pattern recognition tasks is much inferior to that of ANN (Han et al., 2020).
- The second kind of method proposes an approximate error backpropagation based on the spike. With the increase of the timestep, the complexity of SNN training limits the implementation of its deep model (generally 9–11 layers) (Sengupta et al., 2019; Diehl et al., 2015). However, this method lacks strict theoretical analysis at present. In the deep SNN, the gradient is very unstable, and the network has serious degradation problems.
- To extend SNN to the deep layer, the third method proposed is to convert ANN into SNN (ANN-SNN, the schematic diagram of ANN to SNN conversion is shown in Fig. 1). At present, this method is the most effective way to realize the deep SNN (Bu et al., 2022). The ANN-SNN transformation establishes the dynamic relationship between the activation value of the analog neuron and the spiking neuron and maps the parameters of a well-trained ANN to the SNN. Therefore, high-performance SNN can be obtained without additional training (Cao et al., 2014).

Although ANN-SNN is a method to obtain the best performance in current SNN task processing, the performance loss could exist in the converted network (Rathi et al., 2020). The reason for performance loss is that the difference in information processing between ANN and SNN leads to a variety of approximation errors in the conversion process of the network. At present, the existing ANN-SNN methods have large errors when converting deep models, which is far from meeting the requirements of complex task processing. Moreover, the error will gradually accumulate with the increase of network model and the task processing complexity, and the network performance will decline greatly. When the error is mapped to SNN, it is manifested as the change of membrane potential, which affects the spike firing rate of

spiking neurons. Therefore, the spike firing rate of the spiking neuron is the key to affecting the performance of the model.

Based on this, a systematic modeling and deeper analysis of SNN has been conducted in this work. By analyzing the error sources of the ANN-SNN method, the conversion error formula of the deep network was theoretically given. For the conversion error, a TF-reset mechanism and a membrane potential modulation algorithm to reduce the error were proposed. The spike firing rate of the model was guaranteed and the SNN performance with the minimum conversion error was improved.

For clarity, the contributions of this paper are summarized as follows:

- The relationship between spike firing rate and neuronal activation under different reset mechanisms was theoretically analyzed, where the error source of the TF-reset algorithm was also analyzed in detail and the theoretical derivation was given.
- A TF-reset method for spiking neurons was proposed, which effectively reduced the loss of membrane potential of hard reset neurons and solved the problem of over-activation of soft reset method. Based on the analysis of the error source of the reset method, a method of adding modulation membrane potential was proposed to reduce the conversion error of the model.
- The SNN algorithm in this paper achieved better classification accuracy on CIFAR-100 and ImageNet. The model has advantages in energy consumption, network delay, and robustness to noise. Using the method of transfer learning, the SNN-VGG-15 was successfully applied to the classification of remote sensing images and achieved high accuracy.

2. Related work

Neuroscience clarifies the basic principles of brain information processing, and its research purpose is to understand and explore how the nervous system works and completes the corresponding functions (Pozna and Precup, 2012). ANN is an abstraction and simulation of the structure and function (Lee et al., 2016) of biological systems, which is widely used in the field of pattern recognition. However, neurocomputing science explores the mathematical and physical basis of biological systems, which reveals the non-programmed and adaptive information processing capabilities of neural systems (Precup et al., 2021). Compared with biological systems, traditional ANN has its inherent limitations, which are mainly shown in two aspects: (1) the ANN neuron model is too simple to simulate the changes of biological neuron membrane potential and its spike firing process; (2) The ANN model does not use the time information of a single spike internally but processes analog quantities. And neuroscience shows that the biological nervous system processes and transmits information in the form of action potential (i.e., spike). To solve these problems, SNN with more biological authenticity is proposed to solve intelligent application problems in engineering (Tan et al., 2020b). SNN has more realistic modeling and analysis of biological systems based on its unique information coding and processing methods, further deepening the understanding of the basic functions of the nervous system and synaptic plasticity (Boussaid et al., 2017).

At present, ANN-SNN is the most effective way to realize the deep SNN. The SNN obtained by conversion also shows excellent performance on complex image processing tasks (Han et al., 2020; Sengupta et al., 2019; Zhang et al., 2019). In terms of the conversion method, Cao et al. (2014) customized an ANN architecture based on ReLu without bias. However, the disadvantage of their algorithm is that it is necessary to set the spiking neuron threshold for each layer of the network. Based on this, Diehl et al. (2015) realized the automatic determination of the threshold, which is a promotion of Cao et al. (2014). The automatic determination of the threshold depends on the model-based normalization and data-based normalization methods. It can be considered that Diehl's work (Diehl et al., 2015) is the cornerstone of the ANN-SNN

method. To facilitate the conversion, most of the previous work (Bu et al., 2022; Cao et al., 2014) has made certain constraints on the original ANN, such as using the ReLu activation function for neurons, bias 0, and using average pooling instead of maximum pooling. Even so, the performance of SNN suffers a loss of precision compared with the original model. The methods of scaling synapses (Kim and Panda, 2021), adding noise (Sharmin et al., 2020) and constraining synaptic discharge rate (Sengupta et al., 2019) are used to reduce the precision loss of model conversion, but these technologies also complicate the conversion process.

Most of the existing SNN algorithm models (Kheradpisheh et al., 2018; Sengupta et al., 2019; Diehl et al., 2015; Cao et al., 2014) use the hard reset mechanism. Under this mechanism, no matter how much the membrane potential of the neuron exceeds the threshold, the membrane potential will be reset to 0 after it emits a spike. The hard reset mechanism causes the loss of membrane potential, which is also a major source of deep model conversion errors. Therefore, Han et al. (2020) proposed residual membrane potential neurons, which solved the problem of membrane potential loss and improved the spike firing rate. However, we found that the soft reset mechanism has the problem of spiking neuron over-activation. When the input of the sample is large, the membrane potential of the neuron will be many times as the threshold. After the neuron performs the soft reset, its membrane potential still exceeds the threshold. This problem could seriously affect the network's ability to distinguish the input. For the problems of hard reset and soft reset mechanism, the TF-reset mechanism was proposed. The method not only effectively avoids the membrane potential loss of hard reset, but also solves the problem of over activation of soft reset. Finally, a method of modulating the membrane potential was proposed to reduce the cumulative error of network conversion. The SNN based on TF-reset and modulation membrane potential has better performance than hard reset and soft reset. The SNN based on this algorithm is also robust.

3. Theoretical analysis

In this section, we introduce the relevant theories of SNN and analyze the error sources of the network and the reasons for the reduction of network performance. Then, we propose TF-reset and membrane potential modulation algorithm and analyze our algorithm in detail in theory.

3.1. Neuron model

The activation function widely used in ANN is the rectifier linear unit (ReLu) (Xiao et al., 2019). The neurons of ANN process analog information, and b_i^l is a negative value, which can also be processed in the network (b_i^l represents the bias of layer l). SNN deals with discrete spike information (Perez-Carrascothers, 2013; Han and Roy, 2020), which cannot represent negative values. Based on the influence of negative value on SNN, the activation function of ANN used for network conversion removes the bias (Morrison et al., 2007), and the activation value x_i^l (x_i^l represents the activation value of the i th neuron in layer l of the network) of ReLu is written by Eq. (1).

$$x_i^l = \max \left(0, \sum_{j=1}^{m^{l-1}} \omega_{i,j}^l x_j^{l-1} \right), \quad (1)$$

where m^{l-1} represents the number of neurons in layer $l - 1$. $\omega_{i,j}^l$ represents the weight between the neuron i in layer l and the neuron j in layer $l - 1$.

The basic unit of SNN is the spiking neuron (Matsubara, 2017). Integrate-and-Fire (IF) spiking neuron has no leakage and refractory period (Moraitis et al., 2018), and it is also the most commonly used model for ANN-SNN. The membrane potential of the i th IF neuron in

layer l of SNN at time t is denoted by $u_i^l(t)$, which can be calculated by Eq. (2).

$$u_i^l(t) = u_i^l(t-1) + \sum_{j=1}^{m^{l-1}} \omega_{i,j}^l S_j^{l-1}(t), \quad (2)$$

where $S_j^{l-1}(t)$ represents the spike information of the j th neuron in layer $l - 1$ at time t . (If the neuron has a spike at time t , $S_j^{l-1}(t) = 1$, otherwise, $S_j^{l-1}(t) = 0$.) The membrane potential at time t is equal to the sum of the $u_i^l(t-1)$ at time $t - 1$ and the input at the current time.

The input at the current time is represented by $a_i^l(t) = \sum_{j=1}^{m^{l-1}} \omega_{i,j}^l S_j^{l-1}(t)$. When the membrane potential exceeds the threshold, the neuron will emit a spike, and then the neuron will perform the reset operation. The larger the $a_i^l(t)$ is, the higher the spike firing rate of neurons will be. From this point of view, the ReLu (without bias) activation can be approximated to the spike firing rate of IF neurons. This is why ReLu (Sengupta et al., 2019) is selected as the ANN activation function.

3.2. Reset mechanism

The reset mode is an important factor that affects spike firing rate. This section introduces the hard reset methods of SNN, and the TF-reset method was proposed.

3.2.1. Hard reset

The hard reset mechanism (Wu et al., 2019) of spiking neurons is widely used in SNN. Eq. (3) shows the specific operation of this method.

$$u_i^l(t) = \begin{cases} u_i^l(t) & u_i^l(t) < V_{th} \\ 0, S_i^l(t) = 1 & u_i^l(t) \geq V_{th} \end{cases}, \quad (3)$$

where V_{th} represents the threshold of spiking neurons. When the membrane potential $u_i^l(t)$ is less than the threshold V_{th} , the membrane potential will remain unchanged. When the membrane potential exceeds the threshold V_{th} , the membrane potential position will be 0 and $S_i^l(t) = 1$. No matter how much the membrane potential exceeds the threshold, the membrane potential is set to 0. It can be seen that the loss of membrane potential in this method is very large, which reduces the spike firing rate.

3.2.2. TF-reset mechanism

Based on the problem of membrane potential loss of hard reset, the TF-reset mechanism from the aspect of reducing the loss was proposed, which is represented by Eq. (4).

$$u_i^l(t) = \begin{cases} u_i^l(t) & u_i^l(t) < V_{th} \\ u_i^l(t) - \left\lfloor \frac{u_i^l(t-1) + a_i^l(t)}{V_{th}} \right\rfloor V_{th}, S_i^l(t) = 1 & u_i^l(t) \geq V_{th} \end{cases}, \quad (4)$$

where $\left\lfloor \frac{u_i^l(t-1) + a_i^l(t)}{V_{th}} \right\rfloor$ represents the result rounded down. When the membrane potential $u_i^l(t)$ is less than the threshold V_{th} , the membrane potential will remain unchanged. When the membrane potential exceeds the threshold V_{th} , $S_i^l(t) = 1$. At this time, the reset operation of neuronal membrane potential is to subtract $\left\lfloor \frac{u_i^l(t-1) + a_i^l(t)}{V_{th}} \right\rfloor V_{th}$ from the original membrane potential. We order $\eta = \left\lfloor \frac{u_i^l(t-1) + a_i^l(t)}{V_{th}} \right\rfloor$. Since $u_i^l(t) \geq V_{th}$, so $\eta \geq 1$, $0 \leq u_i^l(t) - \eta V_{th} < V_{th}$ can be obtained. From the above inequality relationship, it can be known that the membrane potential after reset is between 0 and V_{th} .

It is assumed that the inputs received by neurons in three consecutive timesteps (t_1, t_2, t_3) are $1.4V_{th}$, $1.2V_{th}$, and $0.4V_{th}$, respectively. Theoretically, the input sum of the neuron in three timesteps is $3V_{th}$, and it is expected to generate three spike signals. In the hard reset method, when the neuron membrane potential is greater than the

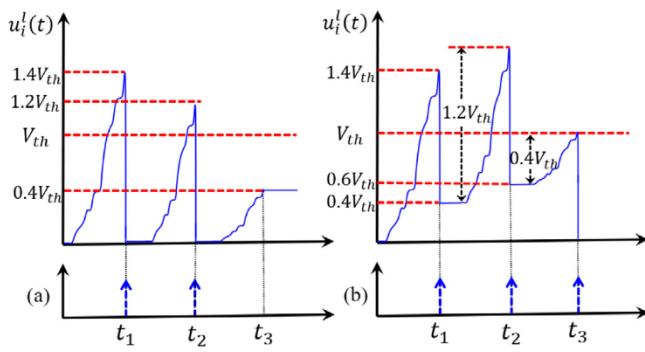


Fig. 2. Schematic diagram of spike firing. (a) Hard reset. (b) TF-reset.

threshold, it will emit a spike, and then the membrane potential will be reset to 0. It can be seen from Fig. 2(a) that the hard reset neurons only have spikes at t_1 and t_2 . Compared with the input, the loss of hard reset membrane potential leads to the loss of spike signal. Fig. 2(b) shows that the TF-reset neuron has a spike signal at t_1 , t_2 , and t_3 respectively. The TF-reset method reduces the loss of membrane potential and ensures the normal emission of spikes.

3.3. Relationship between spike firing rate and activation value

Next, we introduce the membrane potential of neurons in two reset modes and then use it to deduce the relationship between spike firing rate and input. Under different reset mechanisms, the membrane potential $u_i^l(t)$ of spiking neurons at time t can be formally written as Eq. (5).

$$u_i^l(t) = \begin{cases} (u_i^l(t-1) + a_i^l(t)) (1 - S_i^l(t)) & \text{Hard - Reset} \\ u_i^l(t-1) + a_i^l(t) - \eta V_{th} S_i^l(t) & \text{TF - Reset} \end{cases} \quad (5)$$

To ensure the performance of the converted SNN, the spike firing rate of the spiking neuron is approximately proportional to the activation value of the corresponding ANN, $R_i^l \propto x_i^l$ (R_i^l represents the spike firing rate of the i th neuron in layer l of SNN). The length of the SNN spike train is T , the timestep is Δt , and $R_{max} = 1/\Delta t$ (R_{max} represents the maximum spike firing rate of spiking neuron). We can calculate the spike firing rate $R_i^l(t) = \frac{N_i^l(t)}{T}$ at the current time, where $N_i^l(t) = \sum_{t'=1}^T S_i^l(t')$ represents the number of spikes generated by neurons.

We take the first hidden layer of the network as an example for theoretical analysis and deduce the relationship between spike firing rate and neuron activation value (refer to the **Supplementary materials S1** for the specific derivation process). The relationship between spike firing rate and activation value of IF neurons in the first hidden layer under two reset modes is shown in Eq. (6).

$$R_i^l = \begin{cases} x_i^l R_{max} * \frac{V_{th}}{V_{th} + \delta_i^l} - \frac{u_i^l(t)}{V_{th} + \delta_i^l} & \text{Hard - Reset} \\ \frac{1}{\eta} (x_i^l R_{max} - \frac{u_i^l(t)}{t * V_{th}}) & \text{TF - Reset} \end{cases}, \quad (6)$$

where δ_i^l represents the loss of membrane potential when the i th neuron of the first hidden layer is reset. The spike firing rate is not strictly proportional to the activation value of ANN, and there is an approximate error term between R_i^l and x_i^l . Under the hard reset, there is an approximate error caused by residual membrane potential and an additive error term. The TF-reset improves the approximate relationship between them, and there is an additive error term and the reciprocal of the threshold following coefficient.

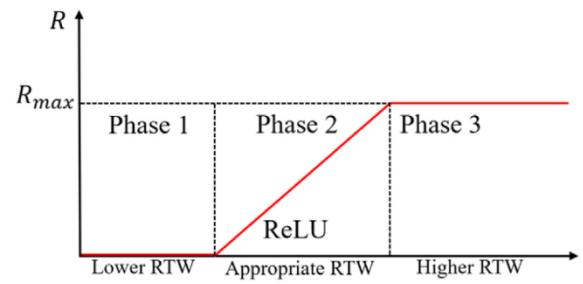


Fig. 3. Relationship between RWT and spike firing rate (IF neurons).

3.4. Analysis of spike firing rate

We define RWT as the ratio of weight to threshold. The spike response curve of the IF neuron is shown in Fig. 3. Smaller RTW (small weight and large threshold) could cause the input of the network to fail to meet the spike firing conditions of neurons (phase 1 in Fig. 3), and larger RTW (large weight and small threshold) may lead to oversaturation of spike firing (phase 3 in Fig. 3). The spike response curves of these two phases are quite different from the ReLu function, and there will be a large approximation error in the process of network conversion. Therefore, it is necessary to reasonably set RTW (also known as threshold balance) to make the spike response curve (IF neurons) in phase 2 correspond to the ReLu function, which can reduce the conversion error caused by the different neuron models.

3.4.1. Spiking firing rate — hard reset mechanism

According to Eq. (6), we analyzed the relationship between RWT and spike firing rate from the perspective of reducing error. The error terms in the hard reset mode are mainly $\frac{V_{th}}{V_{th} + \delta_i^l}$ and $\frac{u_i^l(t)}{V_{th} + \delta_i^l}$. It can be seen from Eq. (2) and supplementary materials that the error caused by the hard reset method can be reduced by increasing the threshold or reducing the weight (RWT reduction). If we reduce the hard reset error term to a very small value, the value of RWT would also be greatly reduced. In this case, the input of the network is not good enough to make the membrane potential over the threshold, and the spike firing rate would be maintained in the first phase of Fig. 3. The error caused by hard reset is inevitable. The increase of threshold and the decrease of weight would reduce the spike firing rate of the network to varying degrees. The decrease of spike firing rate could lead to the sharp reduction of spikes of deep neurons or even no spikes, which is not conducive to the deepening of the network.

3.4.2. Spiking firing rate - TF-reset mechanism

From Eq. (6), the R_i^l of TF-reset can also be formulated as Eq. (7).

$$R_i^l = \frac{x_i^l}{\eta} R_{max} - \frac{1}{\eta} * \frac{u_i^l(t)}{t * V_{th}}, \quad (7)$$

where, the TF-reset method has only one additive error term $\frac{1}{\eta} * \frac{u_i^l(t)}{t * V_{th}}$.

Compared with the hard reset, its proportional relationship is closer to the ReLu function. The relationship between spike firing rate and activation value of the TF-reset mechanism is analyzed in the following three cases:

1. $V_{th} \leq u_i^l(t) < 2V_{th}$, $\eta = 1$, R_i^l can be written as Eq. (8).

$$R_i^l = x_i^l R_{max} - \frac{u_i^l(t)}{t * V_{th}}. \quad (8)$$

In this case, neurons have no loss of membrane potential. Compared with the hard reset, it effectively reduces the information loss caused by residual membrane potential (δ_i^l). Setting RTW reasonably ensures that the spike firing rate of neurons falls in phase 2 of IF response curve (as shown in the curve labeled k in Fig. 4), which is conducive to the deepening of the network.

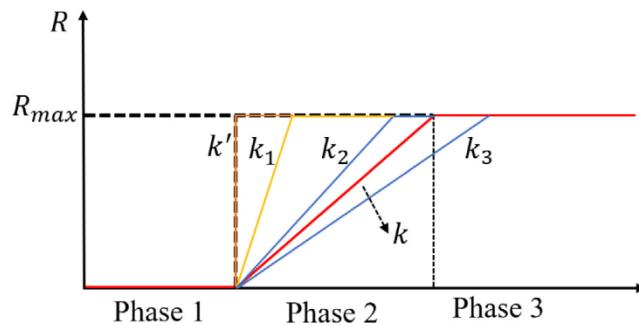


Fig. 4. Schematic diagram of spiking firing rate phase analysis.

2. $u_i^l(t) \geq 2V_{th}$, $\eta \geq 2$. The input received by neurons depends largely on the weight. When the network weight is large, the input x_i^l received by the corresponding neuron would be large. In the hard reset mode, the spike firing rate curve k becomes to k_1 with the increase of input x_i^l (Fig. 4), and R_i^l would be saturated prematurely. In this case, it is difficult for the network to distinguish the input information. In the TF-reset mode, the membrane potential $u_i^l(t)$ will increase with the increase of input x_i^l , and the coefficient η would also increase. In this way, R_i^l is regulated by $\frac{x_i^l}{\eta}$ so that it will not change sharply. The spike response curve k may become to k_2 or k_3 (Fig. 4). Compared with the hard reset, the spike firing of neurons is not greatly affected.
3. Consider an extreme case, $u_i^l(t) \gg 2V_{th}$. The above situation occurs when the input x_i^l is large. In this case, the spike emission of spiking neurons faces the problem of supersaturation. The spike response curve k becomes to k' , which means that no matter how the input information changes, the neuron emits spikes at each timestep (Fig. 4). This causes the network to have no ability to distinguish the input information. In the TF-reset, x_i^l is very large, $u_i^l(t) \gg 2V_{th}$, and $\eta \gg 2$. Under the regulation of coefficient η , $\frac{x_i^l}{\eta}$ has no great fluctuation, so R_i^l is not changed sharply.

From the above analysis, it can be seen that the TF-reset algorithm is more robust than the hard reset method, and it also has a better regulation effect on the spike emission of neurons.

3.5. Spike firing rate of deep spiking neurons

Taking the first hidden layer of the network as an example, the spike firing rate of neurons was analyzed. Under the TF-reset mechanism, we can deduce R_i^l of deep neurons, and its relationship is shown in Eq. (9).

$$R_i^l = \frac{1}{\eta} \left(\sum_{j=1}^{m^{l-1}} \omega_{i,j}^l R_j^{l-1}(t) - \frac{u_i^l(t)}{t * V_{th}} \right). \quad (9)$$

The equation shows that R_i^l is not only related to $R_j^{l-1}(t)$, but also related to the time attenuation approximation error $\frac{u_i^l(t)}{t * V_{th}}$.

This relationship shows that the R of each layer needs to calculate the weighted sum of the approximation error of the previous layer and add its own approximation error. Starting from R_i^l of the first hidden layer, we can formally express R_i^l through recursive expression. The recursive expression is solved iteratively by inserting the expression of the spike firing rate of the previous layer, as shown in Eq. (10).

$$\begin{aligned} R_i^l &= \frac{1}{\eta} (x_i^l R_{max} - \Delta V_{i_l}^l - \sum_{i_{l-1}=1}^{m^{l-1}} \omega_{i_l i_{l-1}}^l \Delta V_{i_{l-1}}^{l-1} - \dots \\ &\quad - \sum_{i_{l-1}=1}^{m^{l-1}} \omega_{i_l i_{l-1}}^l - \dots - \sum_{i_1=1}^{m^1} \omega_{i_2 i_1}^1 \Delta V_{i_1}^1), \end{aligned} \quad (10)$$

where $\Delta V_i^l = u_i^l(t)/(t * V_{th})$. Therefore, the spike rate of the input received by neuron i in layer l is slightly lower, which is based on the quantization error in the previous layer ΔV to reduce. These errors accumulate in higher layers, which explains why it takes longer to achieve a high correlation of network activation and why the spike firing rate of SNN deteriorates in higher layers.

3.6. Spike firing rate modulation

From Eq. (10), the source of deep network errors can be known. To obtain SNN with excellent performance, it would be better to reduce these errors. The error ΔV_i^l can be reduced by increasing t or V_{th} , but both methods increase the delay of the network (O'Connor et al., 2013). Moreover, the increase of t is bound to occupy more computing resources. Simply increasing V_{th} has a great influence on the spike firing of the neuron. Neurons that are difficult to spike will be more difficult to spike when the threshold increases (Hunsberger and Eliasmith, 2015). Therefore, the way of increasing the threshold is also undesirable to a certain extent.

The error ΔV_i^l can also be reduced by reducing the neuron membrane potential $u_i^l(t)$. To ensure that the neurons with insufficient activation are not affected, only the neurons after reset were operated. The membrane potential of the neurons that do not emit spikes remains unchanged. In this way of reducing the error, the adjustment of neuronal membrane potential $u_i^l(t)$ is realized by Eq. (11).

$$u_i^l(t) = \begin{cases} u_i^l(t) & u_i^l(t) < V_{th} \\ \max(u_{rest}, u_i^l(t)' - \phi_u) & u_i^l(t) \geq V_{th} \end{cases}, \quad (11)$$

where ϕ_u represents the modulation potential of neurons. u_{rest} represents the resting potential of neurons ($u_{rest} = 0$). $u_i^l(t)'$ represents the membrane potential after neuron reset. The above equation shows that when the membrane potential of the spiking neuron is lower than the threshold, its membrane potential will remain unchanged. When $u_i^l(t) \geq V_{th}$, the membrane potential $u_i^l(t)'$ will be subtracted ϕ_u ($u_i^l(t) = u_i^l(t)' - \phi_u$). Since SNN cannot represent the negative value, the membrane potential needs to perform $\max(u_{rest}, u_i^l(t)' - \phi_u)$ to ensure that $u_i^l(t)$ is non-negative. This method reduces the cumulative error of the network to a certain extent. If ϕ_u is too large, the spike firing rate will be seriously affected. The appropriate value of ϕ_u should be carefully selected to reduce the cumulative error of the network as much as possible under the condition of ensuring the normal spike firing rate.

4. ANN-SNN

The conversion of ANN to SNN requires a series of conversion operations on the ANN model. It includes model constraints, weight normalization, and spike firing rate modulation.

4.1. Input coding

Compared with ANN, SNN's information processing method is closer to the brain. The spiking neuron processes discrete spike information related to time (Saunders et al., 2018). SNN usually encodes the input information into the discrete spike train based on rate coding (Moraitis et al., 2018). Poisson coding is the most common rate coding method in SNN (Amirsoleimani et al., 2017). Only when the time window of Poisson coding is large enough, it can ensure that the spike input to SNN has a good proportional relationship with the analog value. Moreover, the randomness of spike generation also introduces additional noise. For this reason, repeat coding is used to replace the above method in this experiment (Lu and Sengupta, 2020). Repeat coding directly takes the normalized pixel intensity value as the input of SNN. This method can effectively avoid the introduction of additional noise. The interior of SNN cannot represent the negative value, which is also an

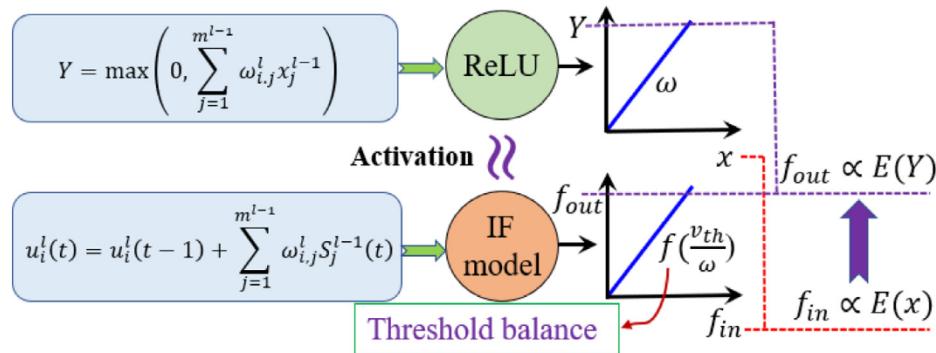


Fig. 5. Mapping relationship between ReLU and IF model.

obvious difference between SNN and ANN. The pixel values in repeat coding should be normalized to the interval [0,1]. In this experiment, the input pixel value is normalized to a range of -1 to 1 so that only the information of the input layer has the negative value, and the information of the subsequent network layer is the discrete positive spike train. Here, the pixel value is normalized to [-1,1], and the spike of the whole network is sparser. This normalization method not only reduces the performance loss of SNN but also saves energy consumption in the model (See **Supplementary material S2** for the analysis and interpretation of the normalization interval).

4.2. Network structure constraints

To ensure the performance of SNN, the internal neural nodes and structure of ANN for conversion need constraints. ReLU was selected as the activation function of ANN neurons and the bias was removed (Yc et al., 2022). In this experiment, IF neurons are selected for SNN, and this model has a good corresponding relationship with ReLU (as shown in Fig. 5, see chapter 3.1 for its analysis). The pre-trained ANN also does not use the batch normalization layer (Neil et al., 2016). Maximum pooling makes the structure of SNN more complex, so SNN usually uses average pooling (Li et al., 2017a) instead of maximum pooling.

4.3. Weight operation

When ANN is converted to SNN, the weight needs to be normalized. The operation of weight is equivalent to the threshold balance of spiking neurons (Kim et al., 2020). The purpose of weight normalization is to keep the spike firing of neurons at a normal level. In this paper, the weight normalization algorithm based on data is used (the schematic diagram is shown in Fig. 6) (Diehl et al., 2015).

This algorithm uses the maximum activation value of neurons in each layer as the scaling factor of weight. The weight of SNN is obtained by multiplying the reciprocal of the maximum activation value by the weight, and the threshold of the spiking neuron is set to 1 (Cao et al., 2014). Note that the algorithm takes the quotient of the maximum activation value divided by the previous layer scaling value as the scaling coefficient of the weight of this layer. After the weight of the previous layer of the network is scaled, the input of the corresponding layer will also be reduced. Therefore, the maximum activation value of this layer will change from the original maximum value to the maximum value divided by the previous scaling value.

4.4. TF-reset mechanism and spike firing rate modulation

The normalized weight is mapped to SNN, and the threshold balance of neurons is preliminarily realized. The reset mode is also an important factor that can affect spike firing rate. The TF-reset method proposed in this paper effectively reduces the loss of neuronal membrane potential. We have derived the cumulative error of deep neurons in Section 3.4.2.

Based on the analysis of the cumulative error, we propose to reduce the error by subtracting a modulation potential ϕ_u from the membrane potential after neuron reset. **Algorithm 1** shows the algorithm flow of the TF-reset mechanism and potential modulation method in SNN.

Algorithm 1: Membrane potential modulation

```

Input: Repeat coding spike train #spike;
Number of Time-Steps #timesteps;
1   net.layer(1).input = spikes
    // Set input of 1st layer equal to spike trains
2   for t ← 1 to timesteps:
3     for l ← 1 to net.layer:
4       for i ← 1 to net.layer.neurons:
5         Equation (2):
6            $V_{neurons}^{i,l,t} = V_{neurons}^{i,l,t-1} + A_i^l$ 
          //  $V_{neurons}^{i,l,t}$  represents the membrane potential of
          // neuron i in layer l at time t.  $A_i^l$  represents the
          // input of neurons at time t.
7           if  $V_{neurons}^{i,l,t} < V_{th}$ :
8             Equation (4):
9                $V_{neurons}^{i,l,t} = V_{neurons}^{i,l,t}$ 
            // The membrane potential remains unchanged
10            else
11              Equation (4):
12                 $V_{neurons}^{i,l,t} = V_{neurons}^{i,l,t} - \text{floor}(V_{neurons}^{i,l,t}/V_{th})$ 
              // Threshold-following reset
13                Equation (11):
14                   $V_{neurons}^{i,l,t} = V_{neurons}^{i,l,t} - \Phi_u$ 
                // Spiking rate modulation
15            end
16        end
17    end

```

4.5. Classification

Biological systems need to identify different objects according to sensory stimuli. These tasks belong to classification problems in machine learning. Biological systems need to recognize different objects according to sensory stimuli. These tasks belong to classification problems in machine learning. The image generally contains the appearance information of the object. Most methods rely on technology based on appearance or features to complete the classification of samples. During model modeling, adding other dimension information based on visual features can improve classification accuracy, such as depth information (Arican and Aydin, 2022), time information (Lee et al., 2016), etc. The analysis method of synthesizing multi-dimensional information is a reasonable and practical classification algorithm.

Weight-normalization base on data:

1. Give trained ANN input test samples
2. Record the maximum activation of each layer, a_{max}^l represents the maximum activation value of layer l
3. ω^l represents the weight of layer l of the network, $\omega^{l'}$ represents the normalized weight.

$$\text{First layer : } \omega^{1'} = \frac{1}{a_{max}^1} \omega^1$$

$$\text{Other layers : } \omega^{l'} = \frac{a_{max}^{l+1}}{a_{max}^l} \omega^l$$

Fig. 6. Schematic diagram: the weight normalization algorithm based on data (Diehl et al., 2015).

Classification is to analyze the input content according to the characteristics for the prediction of sample categories. The classification of the model depends on the classifier. The network in this paper deals with the multi-classification problem. At the last layer of the model, the *Softmax* function is used to predict the category of the input samples. Moreover, the sample categories of the dataset used in this paper are also known. This is different from the K-means (Idb et al., 2022) clustering algorithm, which faces the task of unknown categories. In Section 5.2, the network architecture (as shown in Fig. 8) is introduced, where the output layer is used for model classification. The output layer of the network adopts the full connection, and the number of neurons in the output layer is consistent with the sample category of the dataset. By converting the SNN obtained, we do not set the threshold for the spiking neurons in the output layer but calculate their membrane potential in the time window. The membrane potential $u_i^O(t)$ can be obtained by Eq. (2) ($u_i^O(t)$ represents the membrane potential of the output layer neuron i at time t). During the time window, the membrane potential V_i^O can be calculated by Eq. (12).

$$V_i^O = \sum_{t=1}^T u_i^O(t), \quad (12)$$

where T is the size of the time window. V_i^O represents the membrane potential of the neuron i in the output layer (V_i^O is the membrane potential accumulated in the time window).

The membrane potential of output layer neurons directly can be calculated with *Softmax*, as shown in Eq. (13).

$$y_i = \frac{\exp(V_i^O)}{\sum_{n=1}^{\text{num_class}} \exp(V_n^O)}, \quad (13)$$

where num_class represents the number of neurons in the output layer (i.e., the number of sample categories). y_i is the probability of prediction results. We directly calculate the membrane potential through *Softmax* to get the classification results. This method does not require statistical spikes to speed up the reasoning of the model.

5. Experiments

In the previous section, the error sources of the ANN-SNN method were analyzed. Hard reset and soft reset are the most commonly used reset mechanisms in SNN research. Through analysis, the fact that the hard reset method has serious membrane potential loss has been found. Compared with the hard reset method, the soft reset method avoids the loss of membrane potential, but it has the problem of over-activation of neurons. When the membrane potential is large, the soft reset method

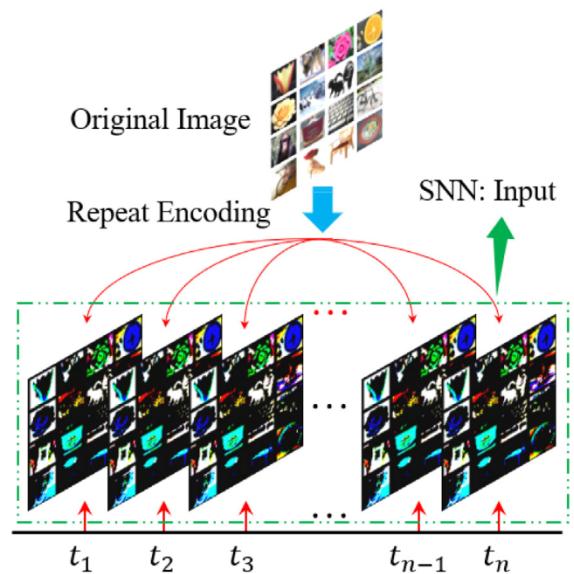


Fig. 7. Schematic diagram of repeat encoding.

will make neurons generate spikes at every moment. That is to say, no matter how much input a neuron receives, it will emit a spike. This may seriously affect the model's ability to distinguish inputs. Aiming at the problems of existing algorithms, our TF-reset mechanism effectively solves the problems of the above algorithms. The conversion error formula of the deep SNN model is also given theoretically. Starting from the error formula, a membrane potential modulation algorithm was proposed to reduce the model conversion error. The membrane potential modulation method is a heuristic algorithm, and we need to get its optimal value from experiments. Next, the performance of the algorithm proposed in this paper will be experimentally verified.

BindsNET (Hazanothers, 2018) is a python software package that can be used for SNN simulation. This paper experiments the conversion from ANN to SNN network based on <https://github.com/BindsNET/bindsnetthreshold-all>. The experiments in this paper are run under the PyTorch framework, using 2 GPU (RTX3090).

5.1. Dataset

5.1.1. CIFAR-100 and ImageNet

In the experiment, the CIFAR-100 and ImageNet were chosen to evaluate the algorithm in this paper. CIFAR-100 contains 60000 (32×32) color images, which include 100 categories in total. 50000 images of the dataset are used for network training and 10000 for testing. ImageNet 2012 is a more challenging task, covering a total of 1000 categories of images. The training set and test set contain 1.28 million and 50000 (224×224) color images respectively. The algorithm is optimized and empirically analyzed on CIFAR-100. The optimization conclusion and related parameters are used in the simulation of ImageNet. The dataset uses Repeat coding to convert the image into the input of SNN. The 16 samples in CIFAR-100 were randomly selected, and Fig. 7 shows the Repeat encoding process of these samples. This coding method directly normalizes the image as the input of each time in the coding time window (time window: t_n).

5.1.2. Remote sensing image scene classification dataset

A. UCM Dataset

The size of UCM dataset images is 256×256 , and their spatial resolution is 0.3 m. It contains 21 scene classes. The dataset has 100 images in each category, with a total of 2100 images. The dataset contains many spatial patterns, which increase the difficulty of remote sensing image classification.

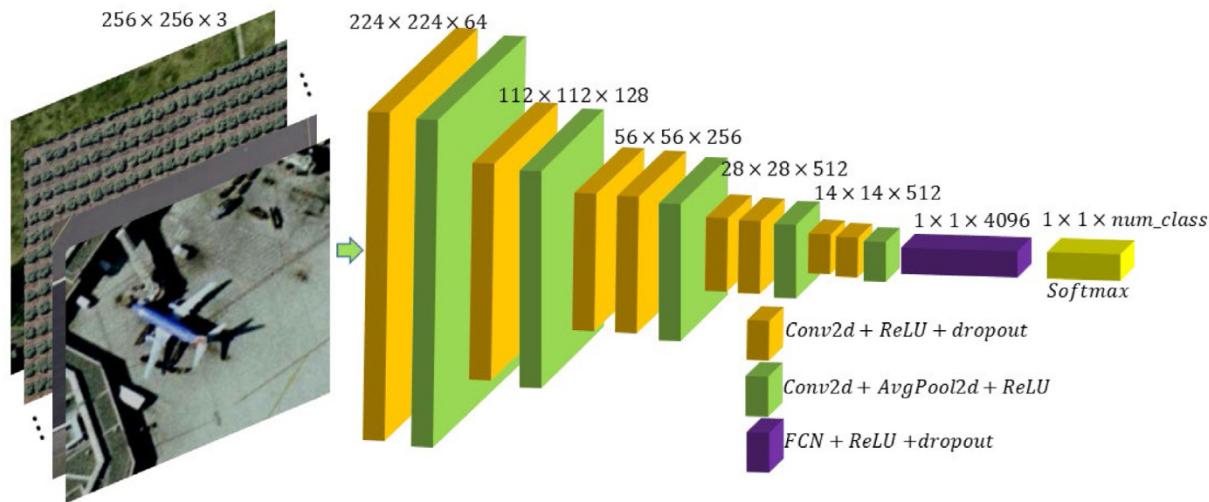


Fig. 8. The network architecture (Input image take UCM as an example).

B. RSSCN7 Dataset

RSSCN7 dataset is a remote sensing image extracted from Google Earth. The size of these remote sensing images is 400×400 . This data set contains seven types of scene images. They are cropped on four different scales, each scale has 100 images, a total of 2800 images.

C. AID Dataset

AID dataset is a remote sensing image dataset, which contains 30 categories of scene images, of which there are about 220–420 images in each category, with a total of 10000 images, of which each pixel size is about 600×600 .

5.2. Network architecture

The ANN architecture to be transformed in this paper follows the standard VGG-16. To reduce the complexity of training and transformation, we remove a linear layer of the standard model and get an improved VGG-15 (The network architecture is shown in Fig. 8). Next, we train the CIFAR-100 on VGG-15 and use the SGD optimizer to set the batchsize, epoch, momentum, and weight delay to 256, 200, 0.9, and 0.0001 respectively. The initial learning rate of training is 0.05. In epoch 81 and epoch 122, the learning rate is divided by 10 respectively. After training, the classification accuracy of the VGG-15 on the CIFAR-100 reached 64.9%. The convergence of the model by the percentage of iterations (the metrics of the convergence: measured as number of iterations, in percentage) was measured. The convergence of VGG-15 on CIFAR-100 is 43%.

Then, we train ImageNet on VGG-15 and use the SGD optimizer to set the batchsize, epoch, momentum, and weight delay to 128, 100, 0.9, and 0.0001 respectively. The initial learning rate of training is 0.01, and the learning rate is divided by 10 every 30 epochs. After training, the classification accuracy on ImageNet reaches 69.05%. The convergence of VGG-15 on ImageNet is 64%. Please refer to Lu and Sengupta (2020) for dataset processing, improved VGG-15 architecture and related parameter settings. Other standard preprocessing techniques used in this work can be found at <https://github.com/NeuroCompLab-psu/SNN-Conversion>.

5.3. Analysis of results

We saved the ANN (VGG-15) fully trained on CIFAR-100 and converted the above model to SNN-VGG-15. Next, the dataset was encoded to realize the simulation of SNN-VGG-15. The input of the SNN model adopts Poisson coding. Because of the randomness of Poisson coding, the SNN model will fluctuate in the results of each test on the same dataset (the fluctuation is small). Therefore, all the experimental results in this paper are the average of the results of five independent experiments.

5.3.1. CIFAR-100

A. Accuracy

The classification accuracy of VGG-15 on CIFAR-100 is 64.9%. The spiking neurons in SNN-VGG-15 are tested by hard reset and TF-reset in turn. The accuracy curve of the model on CIFAR-100 (time window: 512) is shown in Fig. 9(a). The classification accuracy of TF-reset on CIFAR-100 (63.24%) is higher than that of hard reset (62.85%). By comparing the accuracy curve, it can be seen that the SNN with TF-reset converges earlier than hard reset, which means that its network delay is smaller.

Next, we modulate the neuronal membrane potential of the TF-reset and add the modulation potential ϕ_u . The values of ϕ_u were set as $0.1V_{th}$, $0.2V_{th}$, $0.3V_{th}$, $0.4V_{th}$, $0.5V_{th}$ and $0.7V_{th}$. In the experiment, the comparison of soft reset (for the detailed analysis of soft reset, see **Supplementary material S3**) was also showed. The accuracy curve of the SNN-VGG-15 based on different ϕ_u values is shown in Fig. 9(b). In addition, the performance of VGG-15 under different algorithms (the previous algorithms: hard reset (Diehl et al., 2015) and soft reset (Han et al., 2020), our algorithms: TF-reset and modulation potential) was also tested. The model parameters tested in the experiment are consistent. According to the graph, the comparison table of network performance can be obtained (as shown in Table 1, the accuracy value in the table corresponds to the Top-1 value of each network).

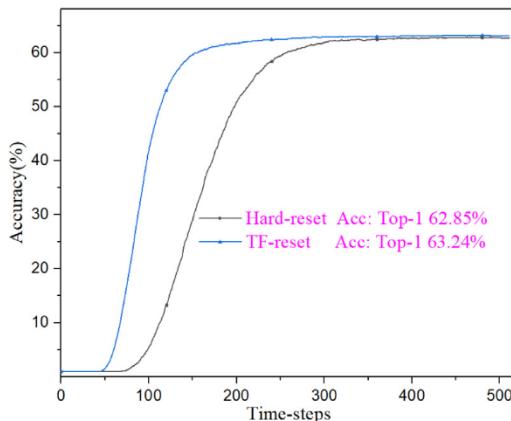
According to Table 1, SNN adopts the TF-reset algorithm, and its accuracy is 63.24%. Compared with hard reset (62.85%) and soft reset (63.09%), SNN using this algorithm has less network conversion loss. To further reduce the conversion error, a modulation potential ϕ_u was proposed. When $\phi_u = 0.3V_{th}$, SNN-VGG-15 achieves the highest accuracy of 64.72% on CIFAR-100. Compared with ANN, its performance loss is only 0.18%, which basically realizes the lossless conversion of the SNN. The timestep in the table represents the size of the encoding time window when the network accuracy reaches 62%. At the same accuracy, the TF-reset has the smallest coding time window, which means the smallest network delay. The smaller the time step required for the model to achieve the same classification accuracy, the lower the energy consumption of its model (see **Supplementary material S4**). The results in the table show that the TF-reset algorithm requires the smallest time step, indicating that the power consumption of our algorithm model is also lower.

B. Spike Firing Rate Analysis

In Fig. 10, the spike firing of neurons in each layer of the network (IFR represents the total number of spikes over the influence time window averaged over the number of neurons) was recorded. In the hard reset mode, the spike signal of the network is the least. Under the

Table 1
Accuracy and network loss of SNN on CIFAR-100.

Model	Reset	ϕ_u	Accuracy (%)	Loss (%)	Time-steps
SNN-VGG-15	Hard reset (Diehl et al., 2015)	/	62.85	2.05	304
SNN-VGG-15	Soft reset (Han et al., 2020)	/	63.09	1.81	223
SNN-VGG-15	TF-reset (This work)	/	63.24	1.66	214
SNN-VGG-15	TF-reset (This work)	$0.1V_{th}$	64.41	0.49	219
SNN-VGG-15	TF-reset (This work)	$0.2V_{th}$	64.57	0.33	248
SNN-VGG-15	TF-reset (This work)	$0.3V_{th}$	64.72	0.18	262
SNN-VGG-15	TF-reset (This work)	$0.4V_{th}$	64.13	0.67	268
SNN-VGG-15	TF-reset (This work)	$0.5V_{th}$	64.39	0.41	273
SNN-VGG-15	TF-reset (This work)	$0.7V_{th}$	64.27	0.53	276



(a) Accuracy curve of hard reset and TF-reset.

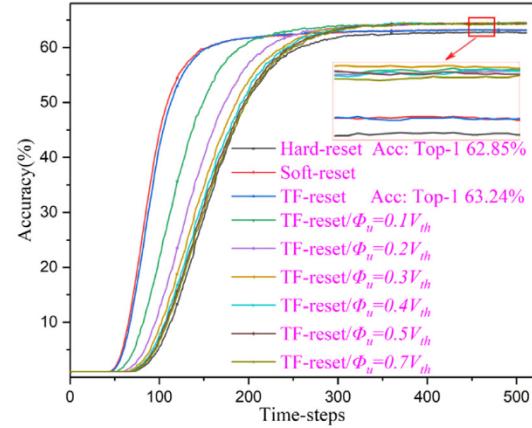
(b) Different ϕ_u accuracy curve of SNN-VGG-15.

Fig. 9. Accuracy curve of the model.

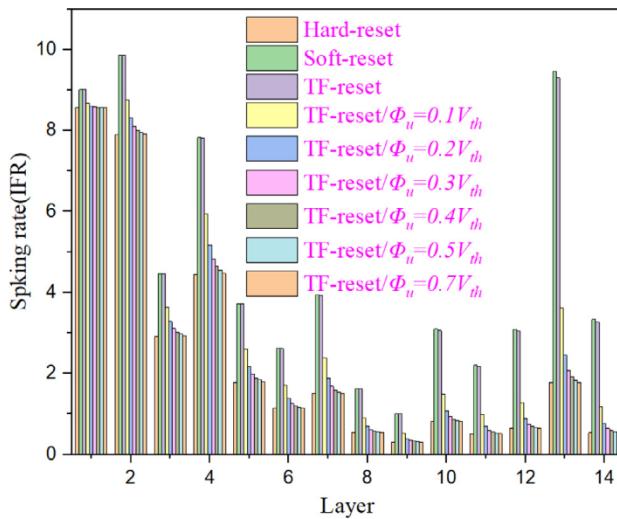


Fig. 10. The spike emission of neurons in each layer of the network.

soft reset mechanism, the network has the most spikes. The number of spikes in the SNN network with the TF-reset is between the two. Therefore, it can be concluded that the TF-reset method reduces the membrane potential loss of hard reset and solves the problem of over activation of soft reset. ϕ_u was added to the model to further reduce the conversion error. Considering the network accuracy and conversion loss, $\phi_u = 0.3V_{th}$ is the optimal modulation parameter. The addition of modulation parameters also makes the spikes in the subsequent network layer of SNN sparser.

C. Energy Consumption Analysis of Model

The energy consumption of the model depends on the total number of spikes in the network. The less the total number of spikes of the model, the lower the power consumption. We define $N_{total-spikes}$ as the average number of spikes generated by a sample in the dataset in the model, and N_i^l as the number of spikes generated by the i th neuron of the layer l . The calculation of $N_{total-spikes}$ can be obtained from Eq. (14).

$$N_{total-spikes} = \frac{\sum_l \sum_i N_i^l}{E_{epoch} \times B_{batchsize}}, \quad (14)$$

where, $B_{batchsize}$ and E_{epoch} represents the batch size and epoch of the model. $E_{epoch} \times B_{batchsize}$ represents the total number of samples tested in the experiment.

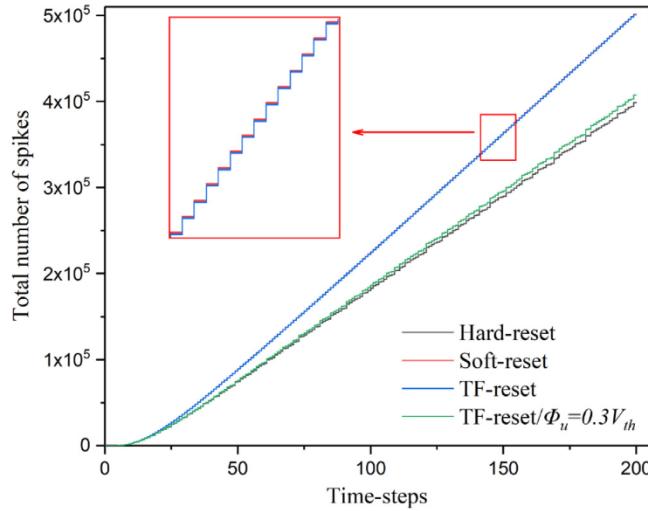
The 100 samples from CIFAR-100 were randomly selected to test the model respectively. The total average number of spikes is shown in Fig. 11. The total number of spikes of SNN based on the hard reset mechanism is the least, which means that its power consumption is the lowest. However, the performance with the hard reset method has a great loss. The soft reset is proposed to solve the membrane potential loss. However, this method will lead to the phenomenon of over-activation of neurons. SNN based on the soft reset has the largest total number of spikes, which means that its power consumption is also the highest. The TF-reset proposed by us not only reduces the membrane potential loss of hard reset but also solves the problem of over-activation of soft reset. It can be seen from the figure that the total number of spikes of the SNN based on the TF-reset is between hard reset and soft reset. Compared with soft reset, the number of spikes of TF-reset is not much reduced.

To reduce the conversion error and energy consumption of SNN, modulation potential ϕ_u was added to the model. In the previous experiments, the model (TF-reset/ $\phi_u = 0.3V_{th}$) achieved the highest accuracy on the dataset and basically realized the lossless transformation of the network. The number of spikes in the model (TF-reset/ $\phi_u = 0.3V_{th}$) is

Table 2

The classification accuracy of the model for different intensity Gaussian noise (CIFAR-100) images.

<i>Sigma</i>	0	1	2	3	4	5	6	7	8	9	10
H-Acc. (%)	62.85	50.51	43.28	40.11	39.04	37.83	36.20	34.20	32.42	30.31	28.28
T-Acc. (%)	64.72	63.80	62.46	61.28	59.79	58.00	56.09	53.78	50.76	47.95	44.98

**Fig. 11.** Total number of spikes in the model.

significantly lower than that in the TF-reset and soft reset models. The power consumption of this method is close to that of the hard reset, but the network conversion loss of this algorithm is significantly lower than that of the hard reset algorithm. The combination of the TF-reset algorithm and modulation potential not only improves the network accuracy but also reduces the power consumption.

D. Gaussian Noise Experiment

Gaussian noise is a common noise in digital images, and its probability density function obeys Gaussian distribution. The characteristic is that there is noise on each pixel of the image, but the depth of the noise is random (any pixel value is superimposed with a random Gaussian value). We randomly select 16 samples in CIFAR-100, and then add Gaussian noise to the samples. The image after adding noise is shown in Fig. 12. *Sigma* represents the standard deviation of Gaussian noise, and the mean value of noise is 0 (the intensity of image noise: $I_{gauss-noise}^{Sigma}$). The larger the standard deviation, the stronger the noise of the image.

This model (TF-reset/ $\phi_u = 0.3V_{th}$) was used to classify Gaussian noise images with different intensities. The experimental results show that the accuracy under different intensities of noise is shown in Table 2 (coding time window: $t_n = 512$). Note that the noise intensity of the image encoded by the input dataset is $t_n * I_{gauss-noise}^{Sigma}$, and the noise intensity input to the model is t_n times that of the original noise graph. That means that when the coding time window is relatively large, even if the noise added to the image is very small, the noise input to the model would be very large.

It can be seen from the results in the table that the classification accuracy of the model decreases with the increase of Gaussian noise standard deviation. When $Sigma = 10$, the classification accuracy of SNN-VGG-15 (hard reset) decreases by about 34%, while the accuracy of SNN-VGG-15 (TF-reset/ $\phi_u = 0.3V_{th}$) decreases by less than 20%. Therefore, compared with the hard reset algorithm, SNN-VGG-15 (TF-reset/ $\phi_u = 0.3V_{th}$) model is robust to Gaussian noise.

E. Contrast

In the above experiments, the TF-reset algorithm and modulation potential on CIFAR-100 were empirically analyzed. Table 3 compares our proposed algorithm with those proposed by predecessors (the results in the table are for CIFAR-100). Through comparison, it can be known that the performance loss of the proposed algorithm is the

Table 3

The algorithm compared with predecessors.

Model	Reset	Loss (%)	Time-steps
XNOR Net (Lu and Sengupta, 2020)	Soft reset	17.74	/
B-SNN (Lu and Sengupta, 2020)	Soft reset	2.73	345
SNN(VGG15) (Lu and Sengupta, 2020)	Hard reset	2.21	362
ResNet-20 (Tan et al., 2020a)	TSC (time-based)	0.54	500
VGG-16 (Tan et al., 2020a)	TSC (time-based)	0.25	500
VGG-* (Yan et al., 2021)	CQ trained SNN	0.40	200
RMP(ResNet-20) (Han et al., 2020)	IF-Hard reset	4.63	2048
RMP(VGG-16) (Han et al., 2020)	IF-Hard reset	0.45	2048
RMP(ResNet-20) (Han et al., 2020)	Soft reset	0.90	2048
RMP(VGG-16) (Han et al., 2020)	Soft reset	0.29	2048
SNN-VGG-15 (This work)	TF-reset/ $\phi_u = 0.3V_{th}$	0.18	262

Table 4

Accuracy and network loss of SNN on ImageNet.

Model	Reset	ϕ_u	Accuracy (%)	Loss (%)
SNN-VGG-15	Hard reset (Diehl et al., 2015)	/	67.76	1.29
SNN-VGG-15	Soft reset (Han et al., 2020)	/	67.66	1.39
SNN-VGG-15	TF-reset (This work)	/	67.68	1.37
SNN-VGG-15	TF-reset (This work)	$0.1V_{th}$	68.07	0.98
SNN-VGG-15	TF-reset (This work)	$0.2V_{th}$	68.14	0.91
SNN-VGG-15	TF-reset (This work)	$0.3V_{th}$	67.98	1.07
SNN-VGG-15	TF-reset (This work)	$0.4V_{th}$	66.05	3.00
SNN-VGG-15	TF-reset (This work)	$0.5V_{th}$	66.02	3.03
SNN-VGG-15	TF-reset (This work)	$0.7V_{th}$	67.79	1.26

smallest compared with previous work, and the coding time window of CIFAR-100 is also relatively small. It can be seen that the SNN based on the TF-reset algorithm and modulation potential not only reduces the performance loss but also greatly reduces its network delay.

5.3.2. ImageNet

In the previous experiment, the ImageNet on VGG-15 was also be trained, and the classification accuracy of the model reached 69.05%. Next, the above conclusion of CIFAR-100 to ImageNet was applied. Firstly, the saved ANN model is transformed to obtain the SNN. Then, the ImageNet is encoded and input to the SNN. The classification accuracy curve of SNN on ImageNet is shown in Fig. 13.

According to Fig. 13, Table 4 was shown, which compares the accuracy and network loss of different models. By comparing the results in the table, it can be known that SNN-VGG-15 (TF-reset/ $\phi_u = 0.2V_{th}$) has achieved the highest accuracy rate of 68.14% in ImageNet, and the loss compared with the original network model is less than 1%. It can be seen that the algorithm in this paper also shows good performance on ImageNet. The models with the highest accuracy on CIFAR-100 and ImageNet are SNN-VGG-15 (TF-reset/ $\phi_u = 0.3V_{th}$) and SNN-VGG-15 (TF-reset/ $\phi_u = 0.2V_{th}$), respectively. The difference between the two models is only the size of modulation potential ϕ_u . It can be seen that the optimal ϕ_u can be flexibly selected for different datasets to achieve the optimal performance of the model.

When the encoding window is 512, the computational complexity of SNN-VGG-15 on CIFAR-100 and ImageNet is $O(337920)$ and $O(1505281)$.



Fig. 12. Gaussian noise image (CIFAR-100).

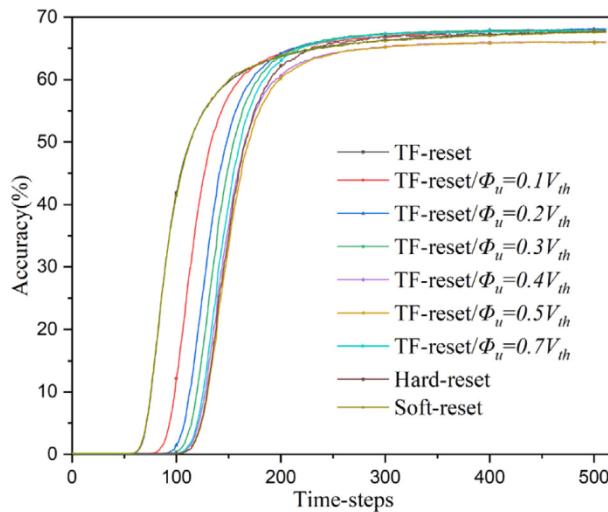


Fig. 13. The test results of SNN models based on different algorithms on ImageNet.

SNN-VGG-15 has an average runtime of 1.56s for testing one CIFAR-100 sample. The average runtime of SNN-VGG-15 for an ImageNet sample is 144.36s. The runtime is measured by the model on a single GPU (RTX3090). The computational complexity of SNN simulations on GPU is higher than that of ANN. Moreover, the runtime of SNN test samples is much longer than that of ANN. SNN combined with a neural morphology platform has great advantages, especially in terms of power consumption. Since this experiment is not supported by the relevant neural morphology platform for the time being, we refer to other articles for a discussion of the cost-effectiveness of the algorithm. [Hu et al. \(2018\)](#) used Intel Stratix 10 TX to run SNN (ResNet) to calculate the energy required to classify ImageNet. The results show that the power consumption of ResNet is more than 9 times that of SNN (ResNet). [Kim et al. \(2020\)](#) running Spiking-YOLO (SNN form of Tiny-YOLO) on TrueNorth chip has a power consumption of about 280 times lower than Tiny-YOLO, and its efficiency is more than 2000 times higher than Tiny-YOLO. By comparing the power consumption of SNN and DNN, it can be seen that SNN combined with a neural morphology platform is more efficient and energy-efficient than traditional models. With the development of neuron morphology chips, SNN can obtain lower energy consumption and higher computing efficiency in the future.

5.4. Remote sensing image scene classification

To verify the effectiveness of the SNN algorithm proposed in this paper for remote sensing image scene classification, the UCM, RSSCN7, and AID datasets were selected to test the model. In the experiment, the method of transfer learning was used. The specific operations of this method are as follows: first, we save the VGG-15 model pre-trained on ImageNet, and then fine-tune the above model according to the size of the remote sensing image. Using the fine-tuning pre-training model to train the remote sensing image dataset, VGG-15 for remote sensing image classification is obtained. Finally, the above model is converted into SNN-VGG-15 by the same conversion method as the previous text for the classification of remote sensing images.

5.4.1. Model training of transfer learning

Compared with other image classification tasks, the samples of remote sensing image datasets are relatively small ([Ghadi et al., 2022](#)). It is easy for the model to fit the remote sensing dataset directly, and the generalization ability of the model will also be reduced. During model training, online data enhancement is used to increase training and test samples to improve the robustness of the model. The data enhancement methods used in the experiment include brightness enhancement, contrast enhancement, horizontal flip, and random direction rotation.

The parameter settings of transfer learning are as follows: the SGD optimizer is used for the training of datasets, and the sizes of epoch, momentum, and weight delay are set to 100, 0.9, and 0.0001 respectively. The initial learning rate of training is 0.001, and the learning rate is divided by 10 after 60 epochs. In the experiment, the dataset obtained through data enhancement is randomly divided into the training set and test set in proportion. The three datasets are divided in the same way, 80% of which are training sets, and the rest are test sets. The accuracy of scene classification of remote sensing images in this paper is the average of five experimental tests. The classification accuracy of the VGG-15 after transfer learning training on UCM, RSSCN7, and AID is 98.82%, 94.32%, and 95.03%, respectively. The convergence of the VGG-15 after transfer learning training on UCM, RSSCN7, and AID is 52%, 73%, and 61%, respectively.

5.4.2. ANN-SNN

VGG-15 has achieved good classification performance on remote sensing datasets by using the method of transfer learning. Based on the ANN-SNN conversion method, VGG-15 is converted into the SNN-VGG-15 model for remote sensing image scene classification. Next, the transformed model was tested. The classification accuracy curve of SNN-VGG-15 on UCM, RSSCN7, and AID is shown in Fig. 14.

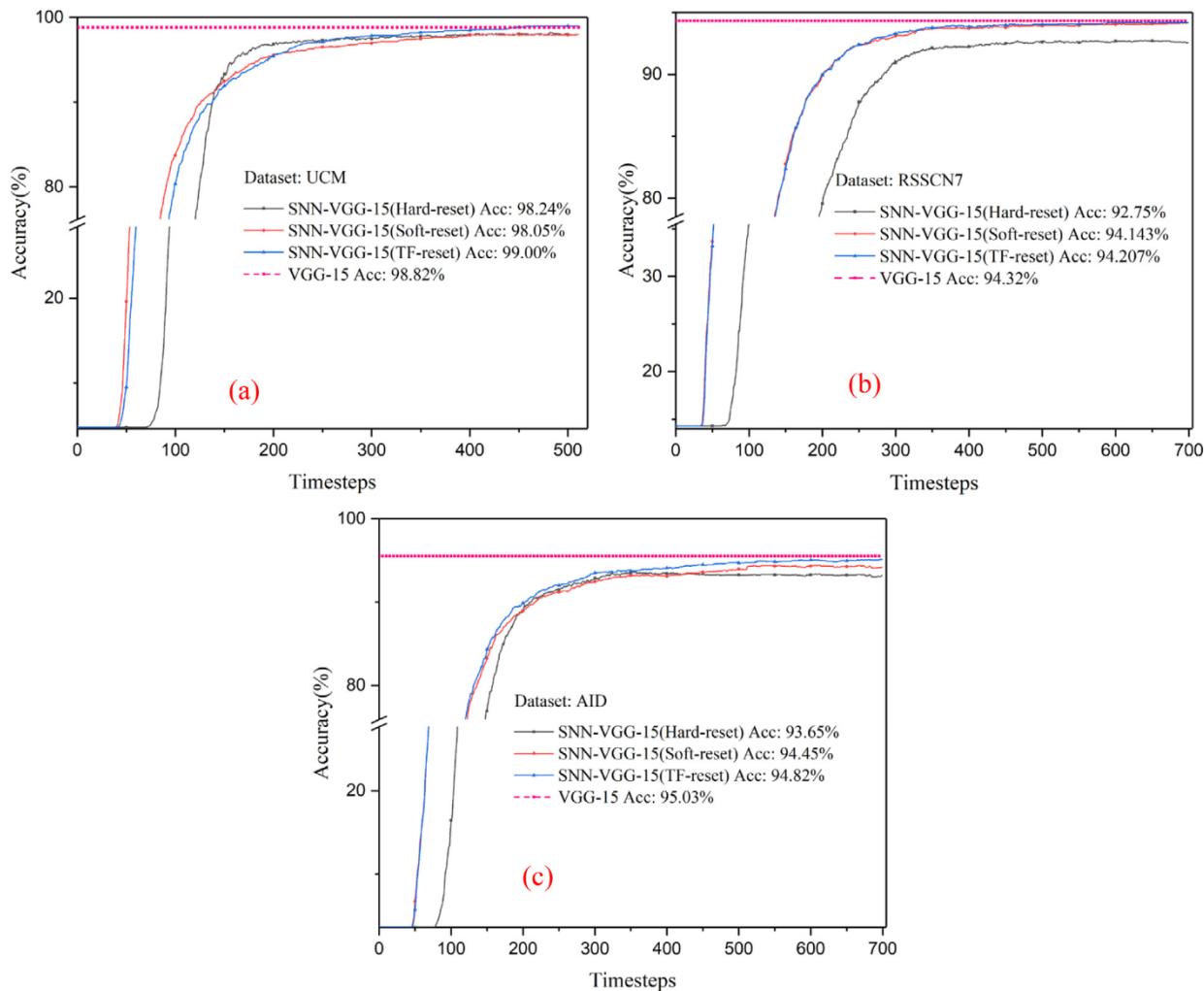


Fig. 14. Graph of scene classification accuracy of remote sensing image.

On UCM, the classification accuracy of SNN-VGG-15 (TF-reset) reaches 99.00%, which is 0.18% higher than that of the VGG-15. The accuracy of SNN-VGG-15 (TF-reset) is 0.76% and 0.95% higher than that of SNN-VGG-15(Hard reset) and SNN-VGG-15 (Soft reset), respectively. SNN-VGG-15(TF-reset) realizes lossless conversion on UCM (as shown in Fig. 14(a)). On RSSCN7, the classification accuracy of SNN-VGG-15 (TF-reset) is 94.207%. Compared with SNN-VGG-15 (Hard reset) and SNN-VGG-15 (Soft reset), the accuracy of SNN-VGG-15 (TF-reset) is improved by 1.457% and 0.064% respectively, while the accuracy is only 0.113% lower than that of the original VGG-15 (as shown in Fig. 14(b)). On AID, the classification accuracy of SNN-VGG-15 (TF-reset) is 94.82%, which is 0.21% lower than that of the original VGG-15 model. The accuracy of SNN-VGG-15 (TF-reset) is 1.17% and 0.37% higher than that of SNN-VGG-15 (Hard reset) and SNN-VGG-15 (Soft reset), respectively (as shown in Fig. 14(c)). Compared with the original VGG-15, the accuracy loss of SNN-VGG-15 (TF-reset) on RSSCN7 and AID is only 0.113% and 0.21%. The algorithm proposed in this paper also basically realizes lossless conversion on RSSCN7 and AID. Experimental results show that compared with hard reset and soft reset methods, SNN-VGG-15 based on TF-reset achieves the highest classification accuracy on three datasets.

5.4.3. Correlation analysis of ANN and SNN

To reduce the performance loss of the model, the ANN-SNN method requires that the activation value of ANN neurons is approximately

proportional to the number of spikes of spiking neurons. Taking the UCM dataset as an example, we analyzed the activation value and spike firing of neurons in the corresponding layers of VGG-15 and SNN-VGG-15 (In the experiment, the SNN-VGG-15(Hard reset) and SNN-VGG-15 (TF-reset) were tested). Fig. 15 shows the relationship between the activation value of ANN neurons in the first two convolution layers of the model and the number of spikes of spiking neurons. To judge the conversion effect of different algorithms more intuitively and accurately, the Pearson correlation coefficient was introduced.

In the experiment, the Pearson correlation coefficient was used to measure the correlation between the neural layers corresponding to ANN and SNN, the correlation of layer l of the model is represented by $\rho_{ANN-SNN}^l$ (It can be calculated by Eq. (15)).

$$\rho_{ANN-SNN}^l = \frac{\sum_{i=1}^{N^l} (a_i^l - \bar{A}_l)(s_i^l - \bar{S}_l)}{\sqrt{\sum_{i=1}^{N^l} (a_i^l - \bar{A}_l)^2} \sqrt{\sum_{i=1}^{N^l} (s_i^l - \bar{S}_l)^2}}, \quad (15)$$

where N^l represents the number of neurons at layer l of the network. a_i^l represents the activation value of neuron i in layer l . \bar{A}_l represents the average value of activation value of layer l neurons. s_i^l represents the number of spikes of spiking neurons i in layer l . \bar{S}_l represents the average value of the number of spikes of layer l neurons.

On the UCM dataset, the correlation between the network layers of the model in hard reset, soft reset, and TF-reset mode was tested,

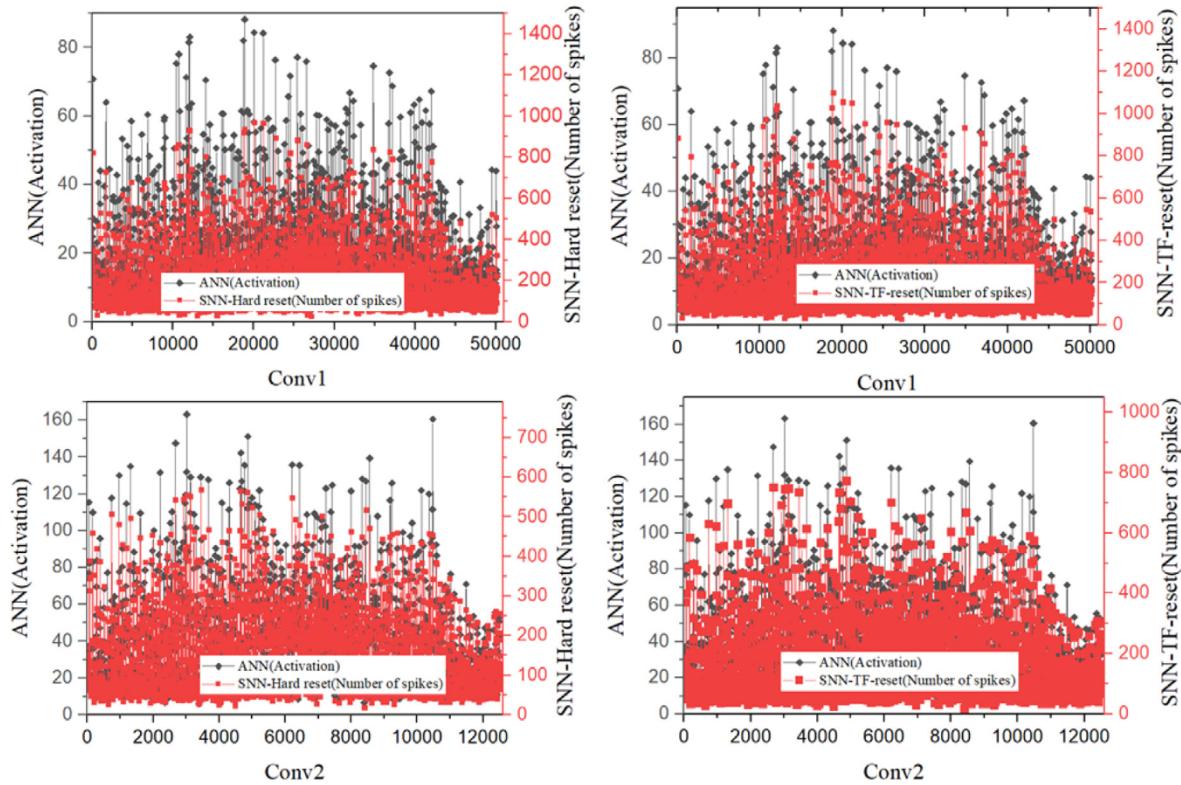


Fig. 15. The relationship between the activation value of ANN neurons and the number of spikes of spiking neurons. (The horizontal axis represents neurons in the convolution layer, the left vertical axis represents the activation value of ANN neurons, and the right vertical axis represents the number of spikes of spiking neurons.)

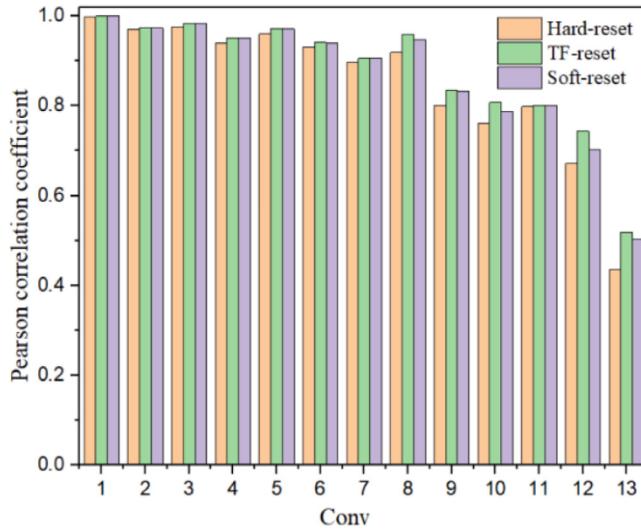


Fig. 16. Correlation between ANN and SNN.

as shown in Fig. 16. In the figure, the horizontal axis represents the convolution layer of the model, and the vertical axis represents the Pearson correlation coefficient between the corresponding convolution layers. From the figure, it can be seen that compared with SNN-VGG-15(Hard reset) and SNN-VGG-15 (Soft reset), the correlation between SNN-VGG-15 (TF-reset) and VGG-15 is the strongest. Therefore, the SNN based on the TF-reset method has a stronger correlation with the original ANN. This correlation experiment shows that our TF-reset algorithm has advantages over hard reset and soft reset algorithms in model transformation.

5.4.4. Membrane potential modulation

The above experiments analyze the classification performance of the TF-reset algorithm on remote sensing image datasets. Next, the modulation potential ϕ_u was added to the above experiment to further improve the performance of the model. The performance of SNN-VGG-15 (TF-reset) with ϕ_u on UCM, RSSCN7, and AID is shown in Fig. 17.

The figure shows the classification accuracy curve of the model on the dataset under the optimal modulation potential value ϕ_u (See Supplementary material S5 for the classification curve under the condition of other modulated membrane potentials (UCM)). The experimental results show that SNN-VGG-15 (TF-reset/ $\phi_u = 0.2V_{th}$) has the best classification performance on UCM, and the accuracy is 99.14%. SNN-VGG-15 (TF-reset/ $\phi_u = 0.3V_{th}$) has the best classification performance on RSSCN7, and the accuracy is 94.54%. SNN-VGG-15 (TF-reset/ $\phi_u = 0.1V_{th}$) has the best classification performance on UCM, and the accuracy is 95.01%. The addition of ϕ_u further improves the classification accuracy of SNN-VGG-15 (TF-reset). The proposed algorithm reduces the cumulative error of the model and realizes lossless transformation.

5.4.5. Contrast

In the above experiments, the classification performance of the VGG-15 model based on transfer learning on remote sensing images was tested. Then, the method based on ANN-SNN successfully realizes the lossless transformation of the model. The SNN in this paper also shows good classification performance in remote sensing image tasks. Next, we list the algorithm of this paper and the work of its predecessors, as shown in Table 5. Through the comparison of the results in the table, it can be seen that VGG-15 and SNN-VGG-15 have advantages in the classification results of the three datasets.

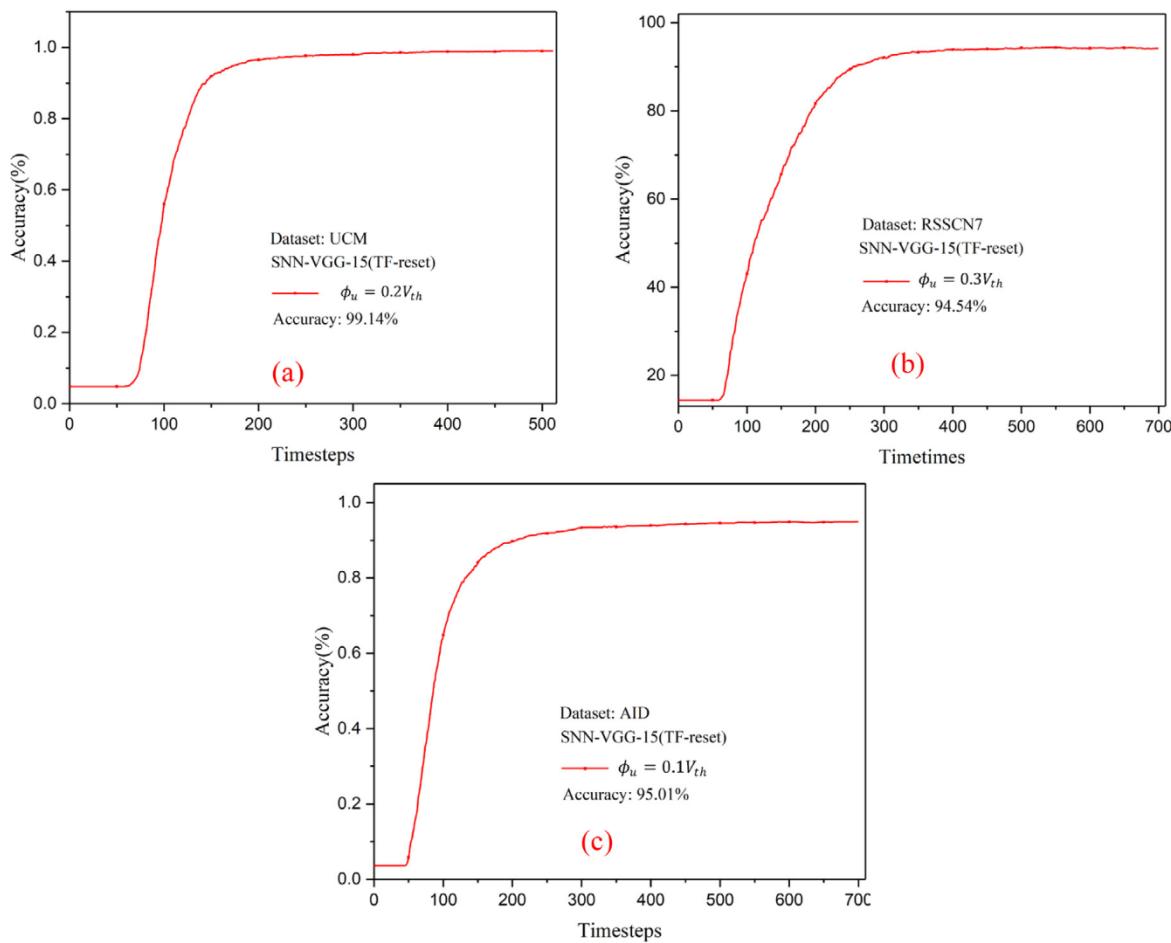


Fig. 17. The performance of SNN-VGG-15 (TF-reset) with ϕ_u .

Table 5
The algorithm compared with predecessors.

UCM dataset		RSSCN7 dataset		AID dataset	
Method	Accuracy	Method	Accuracy	Method	Accuracy
AleNet+MSCP (He and others, 2018)	97.29%	VGG_VD16 (Xia and others, 2017)	87.18%	AleNet+MSCP (He and others, 2018)	92.36%
VGG_VD16+MSCP (He and others, 2018)	98.36%	CaffeNet (Xia and others, 2017)	88.25%	VGG_VD16+MSCP (He and others, 2018)	94.42%
Scenario(I) (Hu et al., 2015)	96.88%	GoogleNet (Xia and others, 2017)	85.84%	DCA (Chaib et al., 2017)	89.71%
Scenario(II) (Hu et al., 2015)	96.90%	DBN (feature) (Qin and others, 2015)	77.00%	Fusion+concatenation (Chaib et al., 2017)	91.87%
DCA (Chaib et al., 2017)	96.90%	TF+HCV (Wu et al., 2016)	86.40%	GoogleNet (Xia and others, 2017)	86.39%
VGG_VD16+IFK (Li et al., 2017b)	98.57%	VGG-S+SVM (Shi et al., 2018)	94.46%	VGG_VD16 (Xia and others, 2017)	89.64%
VGG-15(ours)	98.82%	VGG-15(ours)	94.32%	VGG-15(ours)	95.03%
SNN-VGG-15(TF-reset) (ours)	99.00%	SNN-VGG-15(TF-reset) (ours)	94.207%	SNN-VGG-15(TF-reset) (ours)	94.82%
SNN-VGG-15(TF-reset/ $\phi_u = 0.2V_{th}$)(ours)	99.14%	SNN-VGG-15(TF-reset/ $\phi_u = 0.3V_{th}$)(ours)	94.54%	SNN-VGG-15(TF-reset/ $\phi_u = 0.1V_{th}$)(ours)	95.00%

5.5. Comparison with other SNN models

Currently, SNN training cannot be unified, and there is no universal algorithm. There are two ways to train SNN directly: STDP-based learning algorithm (Kasabov, 2014) and alternative gradient learning algorithm (Tavaneai and Maida, 2019). The direct training algorithm is still limited to shallow network learning and simple task processing. Especially the STDP algorithm, most of its research focuses on the fully connected network. Our algorithm breaks through the limitation of direct training and realizes the application of the deep convolution SNN model in remote sensing scene tasks. The indirect training of SNN is based on the ANN-SNN method, and our algorithm belongs to this category. Although ANN-SNN has achieved good results in the field of pattern recognition (Zhang et al., 2019), its model has performance loss compared with ANN. Diehl et al. (2015) proposed two normalization methods for SNN conversion and achieved success in the four-layer

full-connection network. The use of hard reset of neurons leads to the loss of membrane potential. Our TF-reset method effectively reduces the membrane potential loss of hard reset. The Residual Membrane Potential (RMP) method proposed by Han et al. (2020) successfully realized the conversion of VGG-16 and ResNet-20. However, there is a problem with neuron hyperactivation in RMP. Our algorithm avoids the above problems adaptively.

The above work is to constrain the model (Diehl et al., 2015; Bu et al., 2022) on the ANN. Sengupta et al. (2019) normalized the model on the SNN, and their methods paid more attention to the operation on the SNN network layer. This method needs to operate on the input data, which increases the complexity of conversion. Kim et al. (2020) proposed a channel normalization mechanism to effectively reduce the conversion loss of the model. Han et al. (2020), Sengupta et al. (2019), Diehl et al. (2015) are to operate the model at the level of the network layer. Kim et al. (2020) is to operate the model at the

channel level. Our algorithm is a more fine-grained transformation method than Han et al. (2020), Diehl et al. (2015). The spike firing rate of the model at the neuron level was adjusted to achieve the lossless conversion of the model. Wu et al. (2021) converted SNN (19 convolutions) using multi-bit spike and channel-norm methods and achieved 98.81% classification accuracy in UCM. Although the number of SNN network layers is small, the accuracy of the UCM is 0.33% higher than that of Wu et al. (2021). Compared with the operation on the network layer and channel, our algorithm further reduces the transformation error of the model.

6. Conclusion

Deep learning shows excellent performance in remote sensing image task processing. The improvement of remote sensing image resolution and the increasing complexity of the neural network puts forward higher requirements for the computing power of the computing platform. Due to the limitation of the computing platform, studying a low-power image processing algorithm is a solution to the above problems. SNN model based on event-driven and sparse triggering can provide an ultra-low power alternative for deep learning processing of remote sensing images.

As a network model with biological interpretability, the principle and information processing mechanism of SNN are close to the brain. At present, SNN is also an effective way for deep learning to realize brain-like computing. The TF-reset algorithm and the method of adding modulation potential proposed by us effectively reduce the loss and energy consumption of network conversion, and the algorithm is also robust to noise. The TF-reset method proposed by us cannot completely and truly simulate the spike trigger mechanism of brain cells. The TF-reset is an abstraction of the biological reset mechanism that does not reflect the cell refractory period. At present, our model cannot process complex time information. Although our algorithm has achieved good results in classification tasks, the application of the model in the fields of target tracking and visual navigation cannot meet the requirements.

The advantage of SNN is that if a system has sensors, chips, and powerful SNN algorithms at the same time, its power consumption and running speed are unmatched by DNN. For example, IBM's TrueNorth chip (Akopyan and others, 2015) realizes SNN's real-time recognition of dynamic gestures with extremely low power consumption. Pei and others (2019) published a study on automatic bicycle driving based on the Tianjin chip, which includes SNN's signal processing and fusion. Unmanned bicycles can automatically control balance, recognize voice commands and realize automatic obstacle avoidance. At present, brain-like computing is still in the stage of scientific research. But as far as the existing algorithms are concerned, SNN is still difficult to apply to large practical tasks.

In the subsequent research, we should study algorithms that reflect the real biological mechanism, and further improve the performance and generalization of the model. The study of the high-performance SNN model is also very important for the development of artificial intelligence. The training of SNN is the fundamental reason to limit its development. The following research on the general SNN training algorithm can speed up the application of SNN in the field of remote sensing images.

CRediT authorship contribution statement

Li-Ye Niu: Conceptualization, Methodology, Software, Data curation, Writing – original draft. **Ying Wei:** Investigation, Supervision, Validation. **Yue Liu:** Writing – review & editing.

Declaration of competing interest

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

Data availability

No data was used for the research described in the article.

Acknowledgments

This work is supported by National Nature Science Foundation of China (grant No. 61871106), and the Open Project Program Foundation of the Key Laboratory of Opto-Electronics Information Processing, Chinese Academy of Sciences (OEIP-O-202002).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.engappai.2023.106322>.

References

- Akopyan, F., et al., 2015. TrueNorth: Design and tool flow of a 65 mW 1 million neuron programmable neurosynaptic chip. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* 34 (10), 1537–1557.
- Amirsoleimani, A., Ahmadi, M., Ahmadi, A., 2017. STDP-based unsupervised learning of memristive spiking neural network by morris-lecar model. In: 2017 International Joint Conference on Neural Networks. IJCNN.
- Arican, Erkut, Aydin, Tarkan, 2022. An RGB-D descriptor for object classification. *Sci. Technol. (ROMJIST)* 25 (3–4), 338–349.
- Bing, Z., Meschede, C., Rohrbein, F., Huang, K., Knoll, A.C., 2018. A survey of robotics control based on learning-inspired spiking neural networks. *Front. Neurorobot.* 12, 35.
- Boussaïd, I., Siarry, P., Ahmed-Nacer, M., 2017. A survey on search-based model-driven engineering. *Autom. Softw. Eng.*
- Bu, T., et al., 2022. Optimized potential initialization for low-latency spiking neural networks.
- Cao, Y., Chen, Y., Khosla, D., 2014. Spiking deep convolutional neural networks for energy-efficient object recognition. *Int. J. Comput. Vis.* 113 (1), 54–66.
- Chaih, S., Liu, H., Gu, Y., Yao, H., 2017. Deep feature fusion for VHR remote sensing scene classification. *IEEE Trans. Geosci. Remote Sens.* 55 (8), 4775–4784.
- Diehl, P.U., Neil, D., Binns, J., Cook, M., Liu, S.C., 2015. Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In: International Joint Conference on Neural Networks.
- Esser, S.K., Appuswamy, R., Merolla, P., Arthur, J.V., Modha, D.S., 2015. Backpropagation for energy-efficient neuromorphic computing.
- Ghadi, Y.Y., Rafique, A.A., al Shloul, T., Alsuhibany, S.A., Jalal, A., Park, J., 2022. Robust object categorization and scene classification over remote sensing images via features fusion and fully convolutional network. *Remote Sens.* 14 (7), 1550.
- Han, B., Roy, K., 2020. Deep spiking neural network: Energy efficiency through time based coding. In: Computer Vision – ECCV 2020.
- Han, B., Srinivasan, G., Roy, K., 2020. RMP-SNN: Residual Membrane Potential Neuron for Enabling Deeper High-Accuracy and Low-Latency Spiking Neural Network. *IEEE*.
- Hazanothers, H., 2018. BindsNET: A machine learning-oriented spiking neural networks library in Python. *Front. Neuroinform.* 12.
- He, N., et al., 2018. Remote sensing scene classification using multilayer stacked covariance pooling. *IEEE Trans. Geosci. Remote Sens.*
- Hu, Y., Tang, H., Pan, G., 2018. Spiking deep residual network. 10.48550/arXiv.1805.0135.
- Hu, F., Xia, G.-S., Hu, J., Zhang, L., 2015. Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. *Remote Sens.* 7 (11), 14680–14707.
- Hunsberger, E., Eliasmith, C., 2015. Spiking deep networks with LIF neurons. *Comput. Sci.*
- Idb, A., Rep, A., Abb, A., 2022. Improvement of K-means cluster quality by post processing resulted clusters.
- Kasabov, N.K., 2014. NeuCube: a spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data. *Neural Netw.* 52, 62–76.
- Kheradpisheh, S.R., Ganjtabesh, M., Thorpe, S.J., Masquelier, T., 2018. STDP-based spiking deep convolutional neural networks for object recognition. *Neural Netw.* 99, 56–67.
- Kim, Y., Panda, P., 2021. Revisiting batch normalization for training low-latency deep spiking neural networks from scratch. *Front. Neurosci.* 15, 773954.
- Kim, S., Park, S., Na, B., Yoon, S., 2020. Spiking-YOLO: Spiking neural network for energy-efficient object detection. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 7. pp. 11270–11277.

- Lee, J.H., Delbrück, T., Pfeiffer, M., 2016. Training deep spiking neural networks using backpropagation. *Front. Neurosci.* 10.
- Li, J., Hu, W., Yuan, Y., Huo, H., Fang, T., 2017a. Bio-inspired deep spiking neural network for image classification. In: *Neural Information Processing. Lecture Notes in Comput. Sci.* 294–304.
- Li, E., Xia, J., Du, P., Lin, C., Samat, A., 2017b. Integrating multilayer features of convolutional neural networks for remote sensing scene classification. *IEEE Trans. Geosci. Remote Sens.* 55 (10), 5653–5665.
- Lu, S., Sengupta, A., 2020. Exploring the connection between binary and spiking neural networks. *Front. Neurosci.* 14, 535.
- Matsubara, T., 2017. Spike timing-dependent conduction delay learning model classifying spatio-temporal spike patterns. In: *International Joint Conference on Neural Networks*.
- Moraitis, T., Sebastian, A., Eleftheriou, E., 2018. Spiking neural networks enable two-dimensional neurons and unsupervised multi-timescale learning. In: *2018 International Joint Conference on Neural Networks. IJCNN*.
- Morrison, A., Aertsen, A., Diesmann, M., 2007. Spike-timing-dependent plasticity in balanced random networks. *Neural Comput.* 19 (6), 1437–1467.
- Mozafari, M., Ganjtabesh, M., Nowzari-Dalini, A., Thorpe, S.J., Masquelier, T., 2019. Bio-inspired digit recognition using reward-modulated spike-timing-dependent plasticity in deep convolutional networks. *Pattern Recognit.* 94, 87–95.
- Neftci, E., Das, S., Pedroni, B., Kreutz-Delgado, K., Cauwenberghs, G., 2013. Event-driven contrastive divergence for spiking neuromorphic systems. *Front. Neurosci.* 7 (8), 272.
- Neil, D., Pfeiffer, M., Liu, S.-C., 2016. Learning to be efficient. In: *Presented at the Proceedings of the 31st Annual ACM Symposium on Applied Computing*.
- O'Connor, P., Neil, D., Liu, S.C., Delbrück, T., Pfeiffer, M., 2013. Real-time classification and sensor fusion with a spiking deep belief network. *Front. Neurosci.* 7, 178.
- Pei, J., et al., 2019. Towards artificial general intelligence with hybrid Tianjic chip architecture. *Nature* 572 (7767), 106.
- Perez-Carrascothers, J.A., 2013. Mapping from frame-driven to frame-free event-driven vision systems by low-rate rate coding and coincidence processing-application to feed-forward ConvNets. *IEEE Trans. Pattern Anal. Mach. Intell.* 35 (11), 2706–2719.
- Pfeiffer, M., Pfeil, T., 2018. Deep learning with spiking neurons: Opportunities and challenges. *Front. Neurosci.* 12, 774.
- Pozna, C., Precup, R., 2012. Aspects concerning the observation process modelling in the framework of cognition processes. *Acta Polytech. Hung.* 9 (1), 203–223.
- Precup, R.-E., Bojan-Dragos, C.-A., Hedrea, E.-L., Roman, R.-C., Petriu, E.M., 2021. Evolving fuzzy models of shape memory alloy wire actuators. *Romanian J. Inf. Sci. Technol.* 24 (4), 353–365.
- Qin, Z., et al., 2015. Deep learning based feature selection for remote sensing scene classification. *IEEE Geosci. Remote Sens. Lett.* 12 (11), 1–5.
- Rathi, N., Srinivasan, G., Panda, P., Roy, K., 2020. Enabling deep spiking neural networks with hybrid conversion and spike timing dependent backpropagation.
- Saunders, D.J., Siegelmann, H.T., Kozma, R., Ruszinko, M., 2018. STDP learning of image patches with convolutional spiking neural networks. In: *2018 International Joint Conference on Neural Networks. IJCNN*.
- Sengupta, A., Ye, Y., Wang, R., Liu, C., Roy, K., 2019. Going deeper in spiking neural networks: VGG and residual architectures. *Front. Neurosci.* 13, 95.
- Sharmin, S., Rathi, N., Panda, P., Roy, K., 2020. Inherent adversarial robustness of deep spiking neural networks: effects of discrete input encoding and non-linear activations.
- Shi, Q.P., Li, J., Jiao, Z.C., Wang, Y., Acm, 2018. Deep differential coding for high-resolution remote sensing scene classification. In: *International Conference on Image and Graphics Processing (ICIGP)*, Hong Kong, HONG KONG, 2018. pp. 71–77.
- Taherkhani, A., Belatreche, A., Li, Y., Cosma, G., Maguire, L.P., McGinnity, T.M., 2020. A review of learning in biologically plausible spiking neural networks. *Neural Netw.* 122, 253–272.
- Tan, W., Patel, D., Kozma, R., 2020a. Strategy and benchmark for converting deep Q-networks to event-driven spiking neural networks.
- Tan, C., Šarlija, M., Kasabov, N., 2020b. Spiking neural networks: Background, recent development and the NeuCube architecture. *Neural Process. Lett.* 52 (2), 1675–1701.
- Tavanaei, A., Maida, A., 2019. BP-STDP: Approximating backpropagation using spike timing dependent plasticity. *Neurocomputing* 330, 39–47.
- Wu, J., Chua, Y., Zhang, M., Yang, Q., Li, G., Li, H., 2019. Deep Spiking Neural Network with Spike Count Based Learning Rule. *IEEE*.
- Wu, H., Liu, B., Su, W., et al., 2016. Hierarchical coding vectors for scene level land-use classification. *Remote Sens.* 8 (5), 436.
- Wu, S., Saifei, et al., 2021. Remote sensing imagery scene classification based on spiking neural network. In: *2021 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE.
- Xia, G.-S., et al., 2017. AID: A benchmark data set for performance evaluation of aerial scene classification. *IEEE Trans. Geosci. Remote Sens.* 55 (7), 3965–3981.
- Xiao, R., Yu, Q., Yan, R., Tang, H., 2019. Fast and accurate classification with a multi-spike learning algorithm for spiking neurons. In: *Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19)*.
- Yan, Z., Zhou, J., Wong, W.F., 2021. Near lossless transfer learning for spiking neural networks. In: *Accepted By Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21). Virtual Conference*.
- Yi, A., Ym, A., Ren, F.A., Jx, B., 2022. An adaptive threshold mechanism for accurate and efficient deep spiking convolutional neural networks.
- Zhang, L., Zhou, S., Zhi, T., Du, Z., Chen, Y., 2019. TDSNN: From deep neural networks to deep spike neural networks with temporal-coding. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33. pp. 1319–1326.
- Zhou, S., Li, X., Chen, Y., Chandrasekaran, S.T., Sanyal, A., 2019. Temporal-coded deep spiking neural network with easy training and robust performance.