

# Image restoration based on Spiking Neural Network

Yang Ouyang<sup>1</sup>(31520241154522), Jing Xu(36920241153269)<sup>1</sup>, Shaoqi Yu(36920241153275)<sup>1</sup>,  
Chenxing Lin(36920241153231)<sup>1</sup>

<sup>1</sup>Class AI

## Abstract

Spiking Neural Networks (SNNs) have attracted significant academic interest due to their distinctive features of low power consumption and rapid computing capabilities on neuromorphic chips. However, due to the limited information capacity of spike sequences, SNNs often suffer from significant quantization errors. As a result, most current SNNs are applied to image classification tasks where the precision of output values is less critical. In tasks such as image restoration that require high regression accuracy, SNN are still difficult to handle effectively. In this work, we focus on applying SNN to the field of image restoration, with particular attention to training methods and the design of spiking neuron. Firstly, we improved the hybrid training method to better suit image restoration tasks. Specifically, we use conversion approach to obtain the initial SNN, followed by adding a convolutional layer after the final SNN layer to map discrete spikes to continuous pixel values. We also introduced a membrane potential recycling mechanism to reduce quantization errors, and finally, fine-tuned the model using the surrogate gradient method. Secondly, to further reduce quantization errors, we designed a new spiking neuron, Membrane Potential Reuse Neuron(MPRN). This neuron determines whether to continue firing after network inference based on the residual membrane potential, minimizing quantization errors as much as possible. We conducted experiments on both dehazing and denoising tasks. Experimental results show that our method achieves over 90% model conversion rate within just 10 time steps and decrease energy consumption by 20% to 50% compared to ANNs.

## Introduction

Image restoration is an important task in the field of computer vision, aimed at improving the visual quality and usability of images. Typical image restoration tasks include image dehazing, image denoising, and image super-resolution, among others. In recent years, Artificial Neural Networks (ANNs) have achieved great success in the field of image restoration(Li et al. 2017; Zhang et al. 2017; Liang et al. 2021). However, as network models become increasingly complex, computational complexity and inference energy consumption have become significant challenges for model in the real world(Brown 2020; Rathi et al. 2021).

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Spiking Neural Networks (SNNs) offer a promising solution to the energy consumption problem of ANN models by mimicking the functioning mechanisms of the brain(Maass 1997). SNN transmit information through spike sequences, adding a temporal dimension compared to ANN. At each time step, a neuron will only release a spike if the membrane potential of the spiking neuron exceeds its voltage threshold; otherwise, the neuron remains silent. Additionally, due to the binary nature of spike sequences, costly multiplication operations in network computations can be replaced with more energy-efficient accumulation operations. Therefore, spiking neurons can only trigger sparse additions when they receive spikes; otherwise, they remain idle. This spike-driven SNN demonstrates a significant low-power advantage over ANN when implemented on neuromorphic chips(Yao et al. 2024; Davies et al. 2018; Merolla et al. 2014).

However, due to the limited information representation capacity of spike sequences and the non-differentiability of spikes, training high-performance SNNs has become a challenging problem. Compared to ANNs, which can transmit information using continuous values, SNNs rely solely on 0/1 spike sequences for information transmission. This inevitably introduces quantization errors, leading to a decline in network performance. Through the efforts of researchers, SNNs have achieved accuracy comparable to ANNs in image classification tasks and have significantly reduced energy consumption compared to ANNs(Bu et al. 2022; Lan et al. 2023; Shi, Hao, and Yu 2024), yet research on applying SNNs to image restoration remains limited. With the rising complexity of deep learning-based image restoration models, adapting SNNs for these tasks is essential.

Overall, two challenges lie ahead of us. First, reducing the quantization error in SNNs is critical, as image restoration tasks require higher output precision compared to classification tasks and are more sensitive to quantization errors. Second, the challenge is how to represent discrete spike sequences as continuous image pixel values. To address these challenges, we were inspired by Rathi et al.(Rathi et al. 2019) and proposed a new hybrid training method to better suit image restoration tasks. Specifically, we divide the training process into two stages. In the first stage, we use the ANN2SNN(Cao, Chen, and Khosla 2015; Diehl et al. 2015) method to convert the ANN into an SNN. After the first phase training, we introduced a convolutional layer into

the SNN as a mapper from neuron firing rates to pixel values, thereby expanding the representational capacity of the discrete firing rates. Additionally, we observed that after inference, many neurons in the final layer retain a significant amount of membrane potential. To make full use of this residual membrane potential, we quantize it into an information value and incorporate it into the final result, reducing quantization errors and adding more continuity to the output. Finally, in the second stage of training, we use the direct training method (Neftci, Mostafa, and Zenke 2019; Wu et al. 2018) to train the SNN, further optimizing the model parameters and the pixel mapper. To further reduce quantization errors, we proposed a Membrane Potential Reuse Neuron (MPRN), which determines whether to continue firing based on the current membrane potential after the final time step. This mechanism minimizes information loss by utilizing the remaining potential. The main contributions of this paper are as follows:

- We employed and improved the hybrid training method to train image restoration networks, modifying it to better suit image restoration tasks. To the best of our knowledge, this is the first time that Spiking Neural Networks have been applied to the image dehazing task.
- By analyzing the errors introduced by the conversion method, We propose a Membrane Potential Reuse Neuron (MPRN), which can decide whether to continue spiking based on the membrane potential after SNN inference has finished, in order to reduce quantization errors in the network.
- We evaluated our method on image dehazing and image denoising tasks, and the experimental results show that our approach can achieve excellent results within a small number of time steps.

## Related Work

**Image Restoration.** Currently, deep learning-based image restoration algorithms have been widely applied in tasks. However, these algorithms often rely on complex model computations, which result in significant computational costs, making them challenging to deploy in real-time processing (Su, Xu, and Yin 2022).

**SNN training Method.** Due to the discontinuity of spike sequences, we cannot simply use the backpropagation algorithm to train SNNs in the same way as we do for ANNs. The two most commonly used methods for training SNNs are using surrogate gradient for backpropagation (direct SNN training) (Neftci, Mostafa, and Zenke 2019; Wu et al. 2018) and converting pre-trained ANN models into SNN (ANN2SNN) (Cao, Chen, and Khosla 2015; Diehl et al. 2015). In direct SNN training, backpropagation uses a surrogate gradient by replacing the non-differentiable spiking process with a differentiable function. Using this method, we can train SNNs similarly to ANNs and achieve commendable results in a limited number of time steps. However, this approach performs well only in SNNs with a relatively small number of layers and still shows a significant gap compared to ANNs when applied to complex, large-scale datasets (Lan et al. 2023). Additionally, because SNNs

introduce an extra temporal dimension, their training requires a significant amount of time and consumes substantial GPU resources, leading to high resource demands. The ANN2SNN method involves transferring a pre-trained ANN model to an SNN, allowing for good results without the need for additional training. To achieve better conversion results, the ANN2SNN method typically requires more time steps compared to direct SNN training, meaning it needs longer inference time.

**Image Restoration based on SNN.** SNNs have achieved great success in the field of image classification, and there has been research on their application in tasks such as object detection (Kim et al. 2020; Luo et al. 2024), recognition (Amir et al. 2017; Lan et al. 2023), and segmentation (Patel et al. 2021). However, there is still limited research on their use in the field of image restoration. Song et al. (Song et al. 2024) introduced SNNs to the task of image derain and achieved strong results by designing spiking residual blocks and incorporating a mixed attention mechanism. However, their proposed model is not fully spike-driven, as they incorporated floating-point multiplications during the computation process, which can make it challenging to deploy the model on neuromorphic hardware. Castagnetti et al. (Castagnetti, Pegatoquet, and Miramond 2023) were the first to apply SNNs to the image dehazing task, using a learnable neuron model to achieve excellent results, however, there is still room for improvement.

## Proposed Solution

### Overall Framework

The SNN training framework we propose is shown in Figure 1. The entire training process is divided into two stages. First, we perform the ANN-to-SNN conversion following (Rueckauer et al. 2017) to obtain the initial SNN. Afterward, we replace the neurons in the SNN with a membrane potential reuse neuron (MPRN). Then, we add a Convolutional Pixel Mapper after the output layer of the SNN to achieve a continuous approximation of the discrete firing rates. The residual membrane potential information of the last-layer neurons is computed using the Membrane Potential Recycling Mechanism and incorporated into the final result. Finally, we further optimize the pixel mapper, MPRN, and other network parameters using direct training based on the surrogate gradient method.

### Convolutional Pixel Mapper

In SNNs, discrete spikes are used to transmit information, which inevitably leads to quantization errors when processing continuous image data. Since each neuron can release at most one spike at each time step, the number of spikes emitted by each neuron is  $\sum_1^T s^l(t) \in \{0, \dots, T\}$ . Therefore, the output  $r^L = \frac{\sum_1^T s^L(t)}{T}$  has only  $T + 1$  possible values. This presents a significant challenge for representing continuous pixel values. To achieve finer output, we need to increase the value of  $T$ . For example, to obtain an output value of 0.1, we need at least ten time steps, and to obtain an output value of 0.01, we need at least 100 time steps. But increasing the

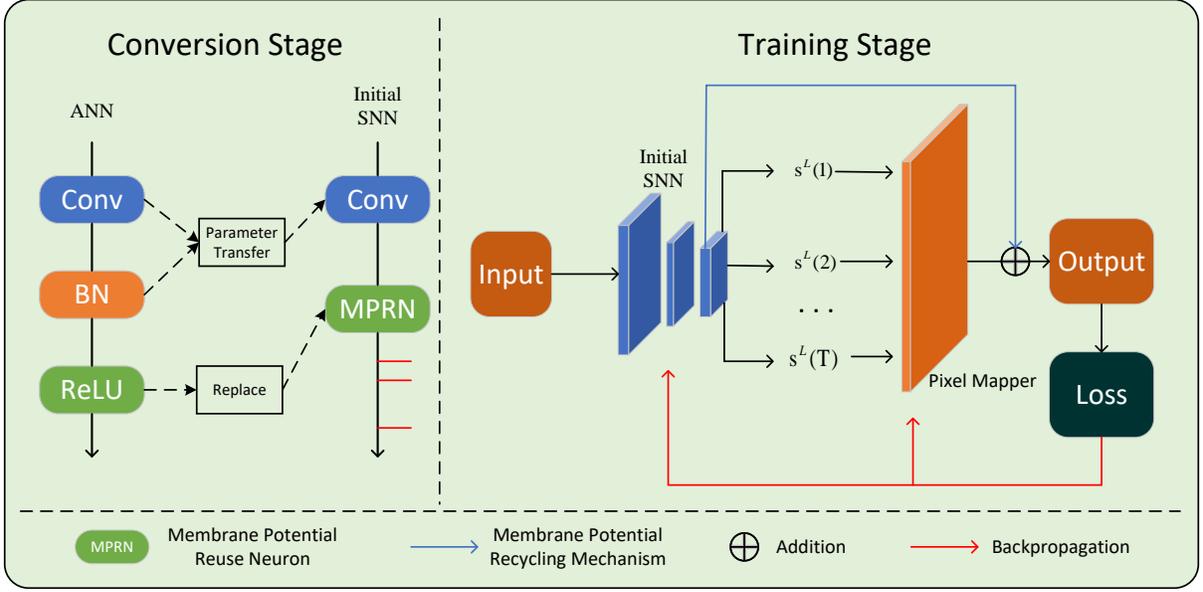


Figure 1: The proposed hybrid training framework. On the left is the conversion process from ANN to SNN, where parameter transfer occurs and ReLU is replaced by MPRN neurons. On the right is the training process, where the Pixel Mapper and membrane potential reuse mechanism are added to the output layer after the initial SNN, followed by backpropagation training using a surrogate gradient function.

value of  $T$  will lengthen the network’s inference time, which is detrimental to practical applications.

To solve this problem, we proposed using a convolutional layer as a mapper from discrete firing rates to continuous pixel values. Specifically, after obtaining the initial SNN in the first stage, we add a convolutional layer after the output layer of the SNN. The discrete spikes from the neuron output are mapped to a more continuous space through the convolution operation. Since the entire mapping process is spike-driven, it only adds a minimal amount of computation.

### Membrane Potential Recycling Mechanism

After the SNN inference ends, since the neuron’s firing cannot convert all of its membrane potential into spikes, there will still be a significant amount of membrane potential remaining on each layer of neurons. This portion of the membrane potential is often discarded either because the time steps are insufficient or because it does not exceed the threshold. In fact, this remaining membrane potential still contains a significant amount of information. For image restoration tasks that require pixel-level precision, it is necessary to reutilize this information to achieve better accuracy. First, let’s analyze the amount of information contained in the residual membrane potential. We assume that a neuron with an initial membrane potential of  $\tilde{v}$ , receiving input  $\tilde{v}$  at each time step, can fully discharge after  $\tilde{T}$  time steps, resulting in a final membrane potential of 0. Therefore, we have the following equation:

$$\frac{\tilde{v} \cdot \tilde{T}}{V_{th}} = n, \quad (1)$$

where  $V_{th}$  is the threshold of the neuron, and  $n$  is the number of spikes released during the entire process. Further, we obtain Eq. 2 .

$$\frac{\tilde{v}}{V_{th}} = \frac{n}{\tilde{T}}, \quad (2)$$

$\frac{n}{\tilde{T}}$  represents the average number of spikes that the neuron can release at each time step. Thus, we can express the membrane potential  $\tilde{v}$  when the threshold is  $V_{th}$  as being equivalent to  $\frac{n}{\tilde{T}}$  spikes. Since we use the firing rate as the output, the amount of information contained in a membrane potential of size  $\tilde{v}$  can be expressed by the following equation:

$$I = \frac{\frac{n}{\tilde{T}}}{T} = \frac{\tilde{v}}{V_{th} \cdot T}, \quad (3)$$

where  $I$  represents the amount of information. To utilize this information, we added an additional data pathway from the last layer of neurons to the output. At the final time step of inference, we calculate the information content from the residual membrane potential of the last layer neurons and add it to the final result. Thus, we obtain the final output of the network as follows:

$$Output = \frac{1}{T} \sum_1^T Mapper(s^L(i)) + I. \quad (4)$$

where the Mapper refers to the pixel mapper mentioned in previous section. We only reutilize the membrane potential of the neurons in the last layer because this approach avoids introducing excessive additional computation and eliminates the need for floating-point multiplication operations. The reutilization of the residual membrane potential from upper layers is achieved through the neuron model proposed in the next section.

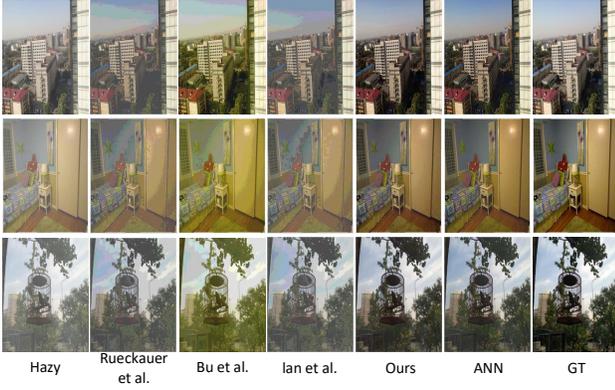


Figure 2: Visual comparisons on SOTS-outdoor and TestSet A. We set the timestep to 10.

### Membrane Potential Reuse Neuron Model

From the analysis in the previous section, we can see that after inference, neurons in the upper layers of the SNN still retain information. Therefore, we propose a new type of neuron designed to reduce information loss in these upper-layer neurons. Starting from the error term  $\left| \frac{v^l(T)}{T \cdot V_{th}} \right|$  according to (Rueckauer et al. 2017), in addition to increasing the number of time steps  $T$ , we can also reduce the magnitude of  $|v^l(T)|$  to minimize the error term. Assuming that after the final time step, the neuron’s membrane potential is  $\tilde{v} > 0$ , we know that firing can reduce the membrane potential; therefore, we can reduce  $\tilde{v}$  by firing, thereby reducing the error. When the  $|\tilde{v}|$  before firing is greater than the  $|\tilde{v}|$  after firing, the error can be reduced. At this point, we have:

$$|\tilde{v}| > |\tilde{v} - V_{th}|, \quad (5)$$

solving this yields:

$$\tilde{v} > \frac{1}{2} \cdot V_{th}, \quad (6)$$

Based on this, we propose Membrane Potential Reuse Neuron Model. After the final time step, we modify the firing equation of the IF neuron according to the following equation:

$$s^l(t) = \begin{cases} \beta, & \text{if } v^l(T+1) \geq \alpha \cdot V_{th}^l, \\ 0, & \text{if } v^l(T+1) < \alpha \cdot V_{th}^l, \end{cases} \quad (7)$$

where  $\alpha$  is a hyperparameter in the range  $(0.5, 1]$ , and  $\beta$  is a trainable parameter. During the given time steps, MPRN follows the firing rules of the IF neuron. After the time steps are exhausted, MPRN extends to a new time step and determines whether it should continue firing based on the aforementioned equation. Additionally, to enhance the adaptability of MPRN, we set the threshold  $V_{th}$  of MPRN as a trainable parameter. This allows both  $\beta$  and  $V_{th}$  to be optimized during the second phase of training. However, there is an issue that needs to be addressed: since the spike value released by MPRN during the extended time step can take on any arbitrary value, this could disrupt the spike-driven nature of the SNN. Fortunately, Guo et al. (Guo et al. 2024) have pointed

Table 1: The energy consumption of the ANN models AOD-Net and DnCNN.

task	dataset	Energy(mJ)
Dehazing	TestSet A	2.467
Denoising	BSD68	393.555

out a solution for us. We can use the reparameterization technique to absorb  $\beta$  into the convolution kernels, ensuring that the spike value remains 0 or 1. Specifically, we can perform reparameterization according to the following equation:

$$G = F * (\beta \cdot B) = (\beta \cdot F) * B = \tilde{F} * B. \quad (8)$$

where  $G$  represents the computation result,  $F$  is the convolution kernel,  $B$  is the 0-1 spike feature map, and  $\tilde{F}$  is the reparameterized convolution kernel.

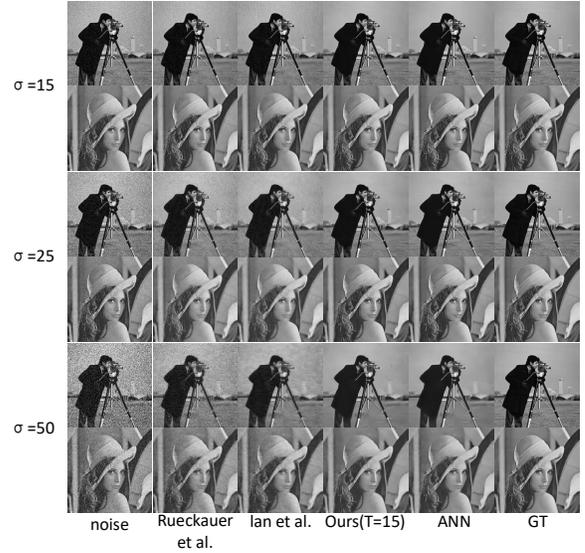


Figure 3: Visual comparisons on denoise task. In methods Rueckauer et al. and lan et al., we set the time step to 20.

## Experiments

### Experimental Setting

**Datasets.** For the image dehazing task, we adopt the same indoor synthetic dataset used in the previous work (Li et al. 2017) for training, and TestSet A for testing, which contains 3,170 hazy indoor images. For the image denoising task, we used the same training and testing sets as (Zhang et al. 2017). The training set consists of 400 grayscale images, while the testing set is BSD68. We evaluated the model’s denoising capability on additive Gaussian noise at noise levels of  $\sigma = [15, 25, 50]$ , where,  $\sigma$  represents the standard deviation of the Gaussian noise.

**Evaluation Metrics.** For denoising and dehazing tasks, we evaluate performance using PSNR and SSIM metrics, furthermore, we retrain some SOTA SNN methods in other

Table 2: The baseline ANN dehazing model uses AOD-Net, with results on the TestSet A datasets. PSNR-loss indicates the loss in PSNR after conversion to SNN, and Rate(%) represents the average of the PSNR conversion rate and SSIM conversion rate.

TestSet A								
Methods	ANN			SNN				
	PSNR $\uparrow$	SSIM $\uparrow$	Timestep $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	PSNR-Loss $\downarrow$	Rate $\uparrow$	Energy(mJ) $\downarrow$
(Rueckauer et al. 2017)	19.53	0.8165	10	16.13	0.6291	3.40	79.8	1.819
			20	18.42	0.7252	1.11	91.6	3.817
			50	19.45	0.7964	0.08	98.6	9.864
(Bu et al. 2022)	18.59	0.6435	5	16.69	0.5926	1.9	90.9	-
			8	18.42	0.7252	1.11	99.4	-
			10	1.54	0.6663	1.05	98.9	-
(Lan et al. 2023)	19.53	0.8165	10	16.44	0.6321	3.09	80.8	-
			20	18.42	0.7252	1.11	91.6	-
			50	19.45	0.7964	0.08	98.6	-
<b>SpikingIR (Ours)</b>	19.53	0.8165	5	19.63	0.7884	-0.10	98.5	1.096
			8	19.76	0.7952	-0.23	99.3	1.898
			10	<b>19.95</b>	<b>0.8106</b>	<b>-0.42</b>	<b>100</b>	2.103

Table 3: Quantitative comparisons BSD68 for image denoising at different noise levels.

BSD68									
$\sigma$	Methods	ANN			SNN				
		PSNR $\uparrow$	SSIM $\uparrow$	Timestep $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	PSNR-Loss $\downarrow$	Rate(%) $\uparrow$	Energy(mJ) $\downarrow$
15	DnSNN	31.730	-	15	31.593	-	0.137	99.6	-
	<b>SpikingIR (Ours)</b>	31.564	0.8930	10	31.378	0.8853	0.186	99.4	192.530
				15	31.467	0.8891	0.097	99.7	262.722
25	DnSNN	29.230	-	15	29.025	-	0.205	99.3	-
	<b>SpikingIR (Ours)</b>	28.879	0.8262	10	28.658	0.8108	0.221	99.2	210.281
				15	28.721	0.8176	<b>0.158</b>	<b>99.5</b>	286.863
50	DnSNN	26.230	-	15	25.94	-	0.290	98.9	-
	<b>SpikingIR (Ours)</b>	25.114	0.6839	10	25.007	0.6649	0.107	99.6	216.248
				15	25.041	0.6725	<b>0.073</b>	<b>99.7</b>	299.949

tasks with the open-source code for fair comparison. Moreover, we calculate the average conversion rate of PSNR and SSIM for the dehazing task and the conversion rate of PSNR for the denoising task to evaluate their performance. The conversion rate is defined as the ratio of the performance of the SNN to the performance of the ANN(Lan et al. 2023).

## Experimental results

The experimental results of our method on dehazing and denoising tasks are shown in Tables 2 and 3. The energy consumption of the two ANN models is shown in Table 1. From the experimental results, we observe that in dehazing task, our method achieves superior performance on both the TestSetA dataset with only 5 timesteps and can achieve loss-less conversion within 10 timesteps. Compared to ANN, our method reduces energy consumption by more than 50% in 5 time steps and by approximately 10% in 10 time steps; compared to other SNN methods, our method can achieve better results in fewer time steps. In the denoising task, we compared our method with DnSNN(Castagnetti, Pegatoquet, and Miramond 2023). Although DnSNN has a higher PSNR, this is actually due to the performance differences in ANN. Com-

paring on PSNR-loss and rate is a more fair choice. It can be observed that our method consistently performs better than DnSNN on these two metrics at the same time steps. Compared to the ANN, our method can reduce energy consumption by nearly 50% at 10 timesteps and by around 10% at 15 timesteps. The visualization results are shown in Figures 2 and 3.

## Conclusion

We propose an efficient SNN conversion method aimed at reducing the significant performance gap between ANN and SNN in image restoration tasks. To achieve this, we mainly optimize the SNN training method and reduce quantization errors. A novel hybrid training approach is proposed, incorporating convolutional pixel mapping and membrane potential reuse mechanisms to reduce network quantization errors. To further enhance error reduction, we propose using MPRN neurons to replace IF neurons, ensuring that the network is pulse-driven while making full use of the residual information in the network. Extensive experimental results on image dehazing and image denoising tasks demonstrate the effectiveness of our proposed method.

## References

- Amir, A.; Taba, B.; Berg, D.; Melano, T.; McKinstry, J.; Di Nolfo, C.; Nayak, T.; Andreopoulos, A.; Garreau, G.; Mendoza, M.; et al. 2017. A low power, fully event-based gesture recognition system. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7243–7252.
- Brown, T. B. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.
- Bu, T.; Fang, W.; Ding, J.; DAI, P.; Yu, Z.; and Huang, T. 2022. Optimal ANN-SNN Conversion for High-accuracy and Ultra-low-latency Spiking Neural Networks. In *International Conference on Learning Representations*.
- Cao, Y.; Chen, Y.; and Khosla, D. 2015. Spiking deep convolutional neural networks for energy-efficient object recognition. *International Journal of Computer Vision*, 113: 54–66.
- Castagnetti, A.; Pegatoquet, A.; and Miramond, B. 2023. SPIDEN: deep Spiking Neural Networks for efficient image denoising. *Frontiers in Neuroscience*, 17: 1224457.
- Davies, M.; Srinivasa, N.; Lin, T.-H.; Chinya, G.; Cao, Y.; Choday, S. H.; Dimou, G.; Joshi, P.; Imam, N.; Jain, S.; et al. 2018. Loihi: A neuromorphic manycore processor with on-chip learning. *Ieee Micro*, 38(1): 82–99.
- Diehl, P. U.; Neil, D.; Binas, J.; Cook, M.; Liu, S.-C.; and Pfeiffer, M. 2015. Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In *2015 International joint conference on neural networks (IJCNN)*, 1–8. ieeec.
- Guo, Y.; Chen, Y.; Liu, X.; Peng, W.; Zhang, Y.; Huang, X.; and Ma, Z. 2024. Ternary spike: Learning ternary spikes for spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 12244–12252.
- Kim, S.; Park, S.; Na, B.; and Yoon, S. 2020. Spiking-yolo: spiking neural network for energy-efficient object detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 11270–11277.
- Lan, Y.; Zhang, Y.; Ma, X.; Qu, Y.; and Fu, Y. 2023. Efficient converted spiking neural network for 3d and 2d classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9211–9220.
- Li, B.; Peng, X.; Wang, Z.; Xu, J.; and Feng, D. 2017. Aodnet: All-in-one dehazing network. In *Proceedings of the IEEE international conference on computer vision*, 4770–4778.
- Liang, J.; Cao, J.; Sun, G.; Zhang, K.; Van Gool, L.; and Timofte, R. 2021. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, 1833–1844.
- Luo, X.; Yao, M.; Chou, Y.; Xu, B.; and Li, G. 2024. Integer-Valued Training and Spike-Driven Inference Spiking Neural Network for High-performance and Energy-efficient Object Detection. *European Conference on Computer Vision*.
- Maass, W. 1997. Networks of spiking neurons: the third generation of neural network models. *Neural networks*, 10(9): 1659–1671.
- Merolla, P. A.; Arthur, J. V.; Alvarez-Icaza, R.; Cassidy, A. S.; Sawada, J.; Akopyan, F.; Jackson, B. L.; Imam, N.; Guo, C.; Nakamura, Y.; et al. 2014. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197): 668–673.
- Neftci, E. O.; Mostafa, H.; and Zenke, F. 2019. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6): 51–63.
- Patel, K.; Hunsberger, E.; Batir, S.; and Eliasmith, C. 2021. A spiking neural network for image segmentation. *arXiv preprint arXiv:2106.08921*.
- Rathi, N.; Agrawal, A.; Lee, C.; Kosta, A. K.; and Roy, K. 2021. Exploring spike-based learning for neuromorphic computing: Prospects and perspectives. In *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 902–907. IEEE.
- Rathi, N.; Srinivasan, G.; Panda, P.; and Roy, K. 2019. Enabling Deep Spiking Neural Networks with Hybrid Conversion and Spike Timing Dependent Backpropagation. In *International Conference on Learning Representations*.
- Rueckauer, B.; Lungu, I.-A.; Hu, Y.; Pfeiffer, M.; and Liu, S.-C. 2017. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Frontiers in neuroscience*, 11: 682.
- Shi, X.; Hao, Z.; and Yu, Z. 2024. SpikingResformer: Bridging ResNet and Vision Transformer in Spiking Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5610–5619.
- Song, T.; Jin, G.; Li, P.; Jiang, K.; Chen, X.; and Jin, J. 2024. Learning a Spiking Neural Network for Efficient Image Denoising. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, 1254–1262. International Joint Conferences on Artificial Intelligence Organization.
- Su, J.; Xu, B.; and Yin, H. 2022. A survey of deep learning approaches to image restoration. *Neurocomputing*, 487: 46–65.
- Wu, Y.; Deng, L.; Li, G.; Zhu, J.; and Shi, L. 2018. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in neuroscience*, 12: 331.
- Yao, M.; Richter, O.; Zhao, G.; Qiao, N.; Xing, Y.; Wang, D.; Hu, T.; Fang, W.; Demirci, T.; De Marchi, M.; et al. 2024. Spike-based dynamic computing with asynchronous sensing-computing neuromorphic chip. *Nature Communications*, 15(1): 4464.
- Zhang, K.; Zuo, W.; Chen, Y.; Meng, D.; and Zhang, L. 2017. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on image processing*, 26(7): 3142–3155.