**Article:**
Xu, Q, Li, Y, Shen, J et al. (4 more authors) (2022) Hierarchical Spiking-Based Model for Efficient Image Classification With Enhanced Feature Extraction and Encoding. IEEE Transactions on Neural Networks and Learning Systems. pp. 1-9. ISSN 2162-237X

https://doi.org/10.1109/tnnls.2022.3232106

# Hierarchical Spiking Based Model for Efficient Image Classification with Enhanced Feature Extraction and Encoding

Qi Xu, Yaxin Li, Jiangrong Shen*, Pingping Zhang, Jian K. Liu, Huajin Tang, and Gang Pan*

***Abstract*—Thanks to their event-driven nature, spiking neural networks (SNNs) are surmised to be great computation-efficient models. The spiking neurons encode beneficial temporal facts and possess excessive anti-noise properties. However, the high quality encoding of spatio-temporal complexity and also its training optimization of SNNs are restricted by means of the contemporary problem, this paper proposes a novel hierarchical event-driven visual device to explore how information transmits and signifies in the retina the usage of biologically manageable mechanisms. This cognitive model is an augmented spiking based framework consisting of the function learning capacity of CNNs with the cognition capability of SNNs. Furthermore, this visual device is modeled in a biological realism way with unsupervised learning rules and advanced spike firing rate encoding methods. We train and test them on some image datasets (MNIST, CIFAR10, and its noisy versions) to show that our mannequin can process greater vital data than present cognitive models. This paper also proposes a novel quantization approach to make the proposed spiking based model more efficient for neuromorphic hardware implementation. The outcomes show this joint CNN-SNN model can reap excessive focus accuracy and get more effective generalization ability.**

***Index Terms*—Hierarchical Structure, Spiking Encoding, Feature Extraction, Spatio-temporal Representations, Noise-immunity.**

## I. Introduction

Pattern recognition task appears in many fields and achieves more and more importance and necessity. Various conventional methods have successfully conducted it, such as kernel regression, Bayesian, and clustering. Primates can recognize the patterns rapidly and precisely [1], [2], [3]. Moreover, human brains have more outstanding performance than computers in intelligent information processing tasks.

How sensory information is processed and transmitted remains a big challenge in the human brain visual systems. Nevertheless, it is strongly supported that the spike train is an optimal way for information representation and transmission [4], [5]. Compared with traditional artificial neural networks (ANNs), Spiking neural networks have shown a more powerful ability for computation on account of their rich neural dynamics which are embedded into spiking neurons. Deriving from the sparse spike sequence, only a few synapses and neurons in the SNN are in an activated status, which enables the SNN to run inferences more efficiently with low computation and power cost. It is competitive for SNN to cope with the high dimensional complexity patterns, by means of the event-driven encoding, training, and dimension reduction mechanisms [6], [7], [8], [9], [10]. The typical SNN is definitely more disadvantageous in feature extraction and coding due to the limitation of the shallow structure. Some deeper and hidden information cannot be captured and extracted by a fully connected layer.

Meanwhile, CSNN [11] and S1C1-SNN [12] implement a biologically plausible way to build a hierarchical cognitive model for the pattern recognition tasks. Both of them use a layer-based feature extractor, compared with deeper and more complex structures, feature extraction and coding capabilities are still limited. Furthermore, the encoding rule embedded into those models is the temporal encoding [13], [14], it is just a linear mapping between features and spikes. This temporal encoding rule is vulnerable in intricate image classification tasks, especially when the images are more complex than handwritten digits. It is still challenging to build robust pattern recognition, which originated from the core representation of sensory stimuli.

The biological spiking neuron could be abstracted as a mathematical model that describes the action potential process of a neuron with rich neural dynamics. Common spiking neuron models include: Hodgkin Huxley Model (HH model) [15], Leaky Integrate-And-Fire Models (LIF model), Spiking Response Model (SRM) [16], etc. LIF model is simplified from HH model. PLIF model is proposed based on LIF which can update the membrane time constant during training. Spiking neurons transmit information by transmitting sparse spike sequence which contains spatio-temporal characteristics.

The conventional spike coding methods of spiking neurons include rate coding, temporal coding, and population coding. Temporal encoding uses timing information of spike firing to encode features such as time-to-first-spike coding, phase coding, rank order coding, latency coding, etc. They are more concerned with the precise timing of spikes. The rate encoding uses the spike firing rate to represent information. It can extract

This work was supported in part by National Key Research and Development Program of China2021ZD0109803), National Natural Science Foundation of China (NSFC No.62206037), Open Research Fund from Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), under Grant No. GML-KF-22-11 and the Fundamental Research Funds for the Central Universities (DUT21RC(3)091). (Corresponding authors: Jiangrong Shen; Gang Pan.)

Qi Xu, Yaxin Li and Pingping Zhang are with the Faculty of Electronic Information and Electrical Engineering, School of Artificial Intelligence, Dalian University of Technology, Dalian 116024, China.

Jiangrong Shen, Huajin Tang, and Gang Pan are with the College of Computer Science and Technology, Zhejiang University, Zhejiang, 310027, China.

Jian K. Liu is with the School of Computing, University of Leeds, Leeds, LS2 9JT, U.K.

features from the number of spikes in a time window. Inter spike intervals (ISI) [17] coding method estimates the average time interval of the spike sequence to encode information. The rate coding does not consider the specific time of spike firing, so it is more anti-interference. Population coding such as gaussian receptive field coding and burst coding uses the neuronal population composed of several neurons to jointly represent information which is more efficient.

In some aforementioned studies, only spatial information is acquired, spatial-temporal information cannot be learned and represented from spike neurons. In these works, the spatial-temporal feature coding mechanism and effective training methods have not been fully explored, which is not only important to achieve fast visual recognition tasks [18], [19], [20], [21] in the neural system, but also robust image classification in harsh conditions [22], [23], [24].

Facing these issues, this work proposes a brain-like event-driven model, combining partial convolutional and pooling components (except the fully-connected layers) and an SNN, the advanced fixed time interval (FTI) and non-fixed time interval spike firing rate encoding methods were embedded into the proposed model. This framework could exploit CNNs' powerful feature extraction and feature-spike encoding capacity of spiking neurons which were integrated into one model.

In this work, we employ the hierarchical model as the basic framework. Additionally, spiking neurons are used as the classifier to make the final classification. We implement unsupervised learning rules to update the model parameters. Since a dynamic arithmetic operation unit can be equivalent to a single neuron [25], [26], we try to define the algebraic transformation of the relationship between features and spikes, these rules are then embedded in the proposed network to convert the features (real value) to corresponding spatial-temporal spike patterns. It is hopeful to enhance the intrinsic representation and the information processing in the brain-like system via these structural and functional units, which are expected to be adaptive for heterogeneous biological neural networks.

Besides, to utilize the energy conservation potential of the spiking based model. This paper also proposed a parameter quantization method to reduce memory and accelerate the computation between spiking neurons. Compared to the real value numerical operation-based spiking models, the quantized model could further exploit the potential when the input and output are spiked. This quantized model is friendly to neuromorphic hardware implementation by reducing the memory and accelerating the information communication between neurons.

## II. Overview of Joint CNN-SNN Model

The visual system is a functional part of the brain. It is found that the external stimuli received by the retina would be encoded as spike patterns via the visual network. Therefore, it can be considered that the information of the visual stimuli transferred from the retina to the brain, which comes from each particular receptive region.

Derived from the structures of vision formation and spike transiting in biological neuroscience, this paper proposed a hierarchical spiking model as shown in Figure 1 to explore further feature extraction and encoding in the human visual system. This model comprises two main parts: a feature extractor and a classifier for decision making. Part of the CNN is used as the feature extractor, which performs as the V1-V4 of visual cortex, and the decision-making part implies the role of the IT (Inferior Temporal) part in the formation of human brain vision. The joint CNN-SNN framework is a unified system model, which is embedded with capacities of feature extraction and encoding information.

### A. CNN based Feature Extractor

We employ a partial CNN as the feature extractor to extract features to imitate the mechanism of information process in the visual sensory system, which is used to capture and filter the image information in the joint CNN-SNN model. The convolutional components in this model were designed for playing the role of the ganglion cells (GCs) parts in the human brain. The GCs, the first layer of the visual cortex, are utilized to acquire information about external stimuli, and then (complex cells) CCs sustain the characteristic dimensions of local areas in the entire image generated by GCs.

The role played by the pooling layer in the joint CNN-SNN model and the CCs layer is similar. A max feature handle for nonlinear operation is applied in pooling layer to fulfill immutability. The Max pooling calculation of different directions, scales, and local positions respectively provide corresponding contrast with scale, reverse and position invariance. This work in [27] accomplished the MAX operation in a biophysically plausible way.

### B. Spiking Firing Rate Encoding Mechanisms

After completing the neuron modeling, the information transmitted by the neuron needs to be encoded. A mainstream method is to transmit the information through the spike firing frequency. In the cerebral cortex, the timing of continuous action potentials is very irregular. One view is that this irregular internal spike interval reflects a random process, so the count of spikes during a specific time window can be estimated by solving the mean value of the response of a large number of neurons. Another view is that this irregular phenomenon may be formed by the precise coincidence of the activity of presynaptic neurons, reflecting a high-bandwidth information transmission pathway. This paper is mainly based on the first point of view, using a random process method to generate a spike sequence.

Images can be encoded as dense spike patterns with the rate based encoding method [28], the firing rate could be expressed as the amount spikes counted during a fixed time window. Dense spikes (Poisson spike trains [29]) are always used by the rate based encoding to stand for the firing rate of neurons.

Given a neural response that consists of a series of spiking kernel functions as described in Eq. (1), $t_i$ is the spike firing time.

$$p(t) = \sum_{n}^{i=1} k(t - t_i), \tag{1}$$

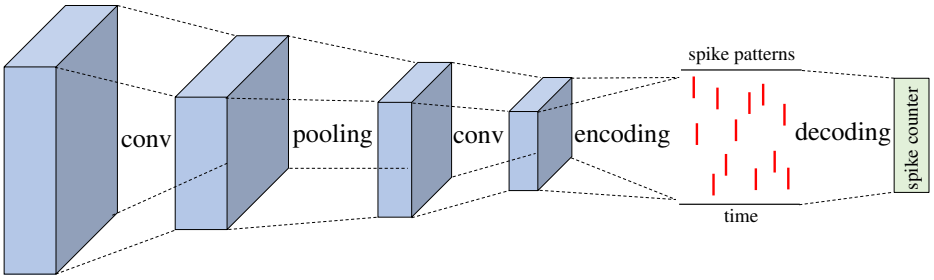



Fig. 1: Visual processing in joint CNN-SNN model. The CNN part imitates the processing of feature extraction from low-level to high-level and a spiking system is used to make the final decision.

$$r(t) = \frac{dn(t)}{dt} = E(p(t)) \tag{2}$$

We can get the whole spikes between time $t_1$ and time $t_2$ as $n = \int_{t_1}^{t_2} p(t)dt$, and the instantaneous spiking firing frequency can be defined as the expectation of the neuron's response function, as shown in the Eq. (2). The average value of the neuron response function in a certain time interval is used as the estimated value of the spike firing frequency as Eq. 3. (There are $M$ spikes fired in the time window $t$)

$$r_M(t) = \frac{1}{M}\sum_{M}^{j=1} p_j(t) \tag{3}$$

Assuming that the spike firing is independent of each other, and each neuron fired certain spikes during a time window. If we suggest there are $k$ spikes fired in a fixed time window, then the $n$ spikes ($n < k$) fired during time $t_1$ to time $t_2$ could be expressed as Eq. (4), in which $p = \frac{t_2 - t_1}{T}$ and $q = 1 - p$.

$$P(n, t_1, t_2) = \frac{k!}{(k-n)!n!} p^n q^{k-n} \tag{4}$$

If $k \to +\infty$, we can get a Poisson distribution based spike train as Eq. (5) showed.

$$p(n, t_1, t_2) = e^{-r\Delta t}\frac{r\Delta t^n}{n!} \tag{5}$$

In this work, we employed two different spike firing rate based encoding methods to encode the outside stimuli to spike trains, one is a fixed time interval (FTI) spike firing rate based method and the other is a non-fixed time interval (NFTI) spike firing rate based method.

In FTI, during the fixed time interval $\Delta t$, the probability of generating a spike signal is $p(n = 1) \approx r\Delta t$. So FTI generates a random number $x[i]$ which conforms to uniform distribution during a fixed $\Delta t$. For every fixed time interval, if $x[i] < r\Delta t$, this neuron fires a spike, otherwise keeps silent as described in eq. 6.

$$s(x[i]) = \begin{cases} 0, & x[i] > r\Delta t, \\ 1, & x[i] < r\Delta t, \end{cases} \tag{6}$$

The other method is NFTI, the probability that the number of spikes emitted during time window $[t_1, t_1 + \tau]$ is 0 can be got as:

$$p(n = 0) = e^{-r\tau} \tag{7}$$

Hence, the probability that the number of spikes emitted during the time window is $p(\tau) = 1 - e^{-r\tau}$, then we can get the probability density distribution of the waiting time between two adjacent spikes as:

$$p(\tau) = \frac{d(1 - e^{-r\tau})}{dt} = re^{-r\tau} \tag{8}$$

Based on Eq. (8), we can get the spike generation way in NFTI method. After a pulse is delivered, a random number is selected to conform to the exponential distribution as the waiting time for the next spike to be delivered.

We show the temporal encoding rule used by S1C1-SNN and CSNN in Eq. (9), $T_{spike}$ denotes the firing time, $T$ is the time window and $A$ represents features of row pixel. This temporal encoding method is too simple to represent the rich spatio-temporal neural dynamics in spiking neurons.

And Eq. (10) demonstrates during a time window $T$ how a pixel is converted into a spike sequence, which is a simplified version of spiking firing rate based encoding method adopted by some works[28], [30]. In a simplified Poisson Distribution, each pixel's value is generally considered as the firing rate $r$. However, this method ignored the important spike time interval as this paper proposed FTI and NFTI did.

Although a large actual cost would impose a greater computational cost, it has excessive fault tolerance. For instance, when we encode an image with Gaussian noise, different encoding regulations (i.e., temporal encoding and sparse encoding) might also switch disparate effects in contrast with rate based encoding, due to the fact this principle maps an actual cost to a spike instruct and mild noises would no longer have an impact on spike patterns drastically.

$$T_{spike} = T - T * A, \tag{9}$$

$$r = \frac{n_{spikes}}{T} = \frac{1}{T}\int_0^T s(t)dt, \tag{10}$$

The joint CNN-SNN model combines the ability of feature extraction from the inputting stimuli and encoding it to a discrete spike based pattern within the proposed spiking firing rate based method FTI and NFTI. The rich neural dynamics which represented by FTI and NFTI spatio-temporal representation can transfer image to sparse spiking patterns, which is consistent with the biological operations in the retina to some extent.

### C. Spiking based Classifier

Considering the spiking module of the proposed joint CNN-SNN, we adopted the classical spiking neuronal model [31] as fundamental units to construct the final readout layer. Due to its strong biological support and effective calculation, we implement the (Leaky Integrate-and-Fire) LIF neuron model to model an SNN. In a LIF neuron model, the membrane potential $V$ could be demonstrated in Eq. (11) and Eq. (12)

$$C_m \frac{dV}{dt} = g_l(E_l - V) + I, \tag{11}$$

$$V = V_{rest}, \quad if\ V \geq V_{th}, \tag{12}$$

$C_m$ stands for the membrane capacitance, $g_l$, $E_l$, and $I$ means conductance, equilibrium leakage potential, and total input current, respectively. All inputting pictures can be encoded through weighted synapses and presented by a LIF neuron's membrane potential $V(t)$.

This paper adopted an unsupervised learning rule to train the spiking classifier. Based on a more formal and detailed description, there is an increase in synapses from the repetitive and continuous operation between the pre-and post-synaptic neuron. Spiking Timing Dependent Plasticity (STDP) is one of the widely used learning rules for modeling spiking models: considering a pair of related units (A and B), if the pre-neuron A fires before the post-neuron B fires, we can say that the firing of B is associated with the firing of A, which leads to a result that the synapse enhances among two neurons increases, this is defined as the long-term potentiation (LTP); Instead, it is named long-term depression (LTD).

$$\Delta W_{ij} = \begin{cases} M_+ exp(\frac{t_j - t_i}{\tau^+}), & if\ t_j < t_i\ (LTP), \\ M_- exp(\frac{t_i - t_j}{\tau^-}), & if\ t_j > t_i\ (LTD), \end{cases} \tag{13}$$

Eq. (13) described the STDP rule in detail, where the range of pre-post synaptic intervals belonging to wakening (LTD) or synaptic strengthening (LTP) can be determined by $\tau^+$ and $\tau^-$. $M_+$ and $M_-$ represent the learning rates, which can decide the maximum numbers of synaptic alterations of LTP and LTD respectively.

## III. Experimental Results

In this part, we employ three benchmark datasets (basic MNIST [32], background MNIST [33] and background-random MNIST [33]) to evaluate the efficiency of joint CNN-SNN model, they are shown in Figure 2. Each dataset comprises ranging from 0 to 9 and size as 28×28 gray-scale images of digits. Each dataset is spilt into the training set (50,000 samples) and the test set (10,000 samples).

Besides, most current studies focused on the MNIST image classification tasks, compared to MNIST dataset, classifying the accurate images on some more pixel-level complex datasets such as CIFAR10 is more challenging to current SNN structures [34]. CNN-based feature extracting method could get richer information from such natural visual scenes compared with typical SNN models. CIFAR10 dataset consists of 32×32 colorful RGB-based images in 10 categories, which is significantly different from grayscale images in MNIST.

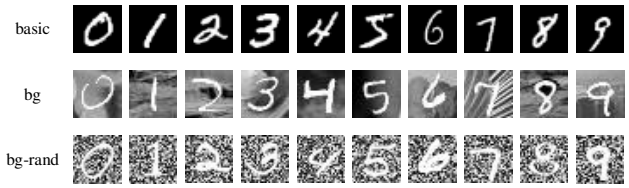



Fig. 2: MNIST and its noisy versions.

We split the original CIFAR10 dataset into two subsets: training and test sets (50,000 versus 10,000 samples). This paper added controllable noise in CIFAR10 to generate three variations as shown in Figure 3, we used $\sigma$ to adjust the noise intensity. $\sigma = 0$ means clear CIFAR10 images, and the noise intensity is increasing when $\sigma$ becoming larger. In this paper, we adopted standard stochastic gradient descent (SGD) with momentum (0.95) for the training CNN part, and set different training hyperparameters such as training epochs (300, 400), learning rate (0.1, 0.08) and different batchsize from 10 to 128 respectively for MNIST and CIFAR10 CNN backbone training.

To demonstrate the networks' generation capacity to the noisy images (i.e., variations of MNIST and CIFAR10), we slice different amounts of the training and test sets to certify that the joint CNN-SNN model can achieve better performance than that of other cognitive models on the small-scale training set.

The training approach of this joint model is divided into two stages: First, we use the Stochastic Gradient Descent (SGD) algorithm as the optimizer to optimize the parameters of a whole CNN. Then, the SNN part of the joint CNN-SNN model, as the final classifier, is trained with the unsupervised STDP rule.

### A. Experimental Settings

We employ two-processor NVidia GeForce GTX 1080Ti GPUs and Intel(R) Xeon(R) Core CPU to conduct all experiments. The soft operating system is Ubuntu 16.04. Tensorflow [35] and Brian [36] are applied to optimize and validate the proposed joint CNN-SNN model.

In terms of MNIST and CIFAR10, we trained two different CNNs and implemented its convolutional and pooling layers as to extract features. Their architectures are 6C6@28×28-12C5-24C5-P for MNIST and 32C5@32×32P232C5@16×16P2-64C5@8×8P2 for CIFAR10 respectively. The SNN architecture is analogous with this framework [37], and we adjust the SNN size for suiting these two different CNN structures. Excitatory and inhibitory neurons are connected with each to each pattern and every inhibitory neuron is related to all excitatory neurons. This kind of structure could present and mimic lateral inhibition and result in opposition amongst excitatory neurons.

### B. Comparison of Spike Firing Rate Methods Between FTI and NFTI

In order to further show the difference between fixed time interval (FTI) and Non-fixed time interval (NFTI) spike firing


Fig. 3: CIFAR10 and its variations with different noise intensities.

rated encoding, FTI and NFTI are embedded into the proposed model. And the experiments were evaluated on MNIST and its variations. Further, we adopt clean MNIST images as training datasets, and the amount of training images is from 500 to 10000.

Figure 4 shows the comparison between FTI and NFTI. From this table, we can see that there is not much difference between the FTI and NFTI in general. With the increase in the number of training images, the classification performance of the NFTI model is slightly higher than that of FTI, especially when we use 10000 basic pictures for training and 2000 basic pictures for testing, the accuracy is the best, reaching 90.2%. Whether with FTI and NFTI, they both did not behave well on noisy test datasets, although the overall trend is upward. The classification results from noisy test datasets including background (bg) noise and background-random (bg-rand) noise indicate that although CNN could extract rich information from images, it does not work when it is dealing with too intensive noise such as bg and bg-rand.

To further explore the neural dynamics from spiking based models on more complex images, we compare our joint CNN-SNN model with NFTI encoding method with the other two spiking based networks CSNN [11] and S1C1-SNN [12]. All these spiking based models are evaluated both on MNIST CIFAR10 and its variations.

S1C1-SNN is a basic SNN model that used fixed manual feature extraction. CSNN is an advanced SNN model that was trained through supervised learning rules and utilized linear temporal encoding as its feature-spike mapping rule. Since they both used SNN as a classifier, the experimental setting is the same with mixed training-test images which could show the generalization ability of spiking based models. For MNSIT, we trained the systems on clean MNIST images (basic) and tested on its noisy versions basic, bg and bg-rand MNIST. For CIFAR10, we train the models on different noise intensity levels in CIFAR10 images and test on their corresponding test datasets. These conditions are designed for evaluating the generalization ability handling to different intensity noise.

We conclude in Table I all of the test accuracies of MNIST and CIFAR10. From the left are the classification accuracy of S1C1-SNN, CSNN, and the model we propose. As shown in this table, when we adopt the clean (basic) as the training set, the joint CNN-SNN could achieve significantly better test accuracies than that of the other two models. For instance, when the training and test sets are the basic MNIST and its corresponding test set, the joint CNN-SNN achieves approximately 86.5%, which is not affected by the size of the dataset, while CSNN and S1C1-SNN can only reach roughly 85% and 77%, which is obviously influence with the changes of the number of training samples.

Two different instances were trained on the clean datasets and tested on noisy datasets to record the comparable performance, through a way of and massive the joint CNN-SNN behaves worse (less than 30%) than the two different networks. Since the other two models were trained on the supervised learning rule the Tempotron, compared with the unsupervised method adopted by this model, supervised learning can obtain image labels in advance, thereby improving the efficiency of the training process.

As for training on CIFAR10 and its variations with different intensity Gaussian noise, the joint CNN-SNN demonstrates larger performance advantages compared to the other two models. Table I shows that when the amount of training images is restricted, the proposed joint CNN-SNN shows better performance than S1C1-SNN and CSNN. For instance, when 500 clean CIFAR10 training samples are used and tested on basic MNIST, the accuracy of the joint CNN-SNN is 58.0%. By contrast, the accuracies from the other two models are only 52% and 54% on clean CIFAR10 images. In addition, as the number of training samples increases. the gap is shrinking. When we use the bg MNIST as the the training set, three models achieve almost the same performance.

When the proposed model was trained on different densely noisy images (represented by different $\sigma$), the proposed model still performs more robustly and better than the other two models. On the one hand, the encoding mechanisms adopted by S1C1-SNN and CSNN models are only simple linear feature-spike transfer rules and may not be able to give full play to the advantages of neural dynamics, especially when compared with rated based encoding rules of joint CNN-SNN model, two models behave poorly in noisy environments. On the other hand, the proposed joint CNN-SNN achieves better performance owe to the more deeper and reasonable structure than S1C1-SNN and CSNN. These evidences also prove importance of appropriate deep structure. When the training images increasing, the corresponding classification accuracies become better but not significant. The proposed model shows a sharp decrease from 73.5% to 50.3% with the noise intensity reaching to peak at $\sigma = 0.1$. S1C1-SNN and CSNN also behave worse at this situation which could prove that classification tasks on CIFAR10 is much more difficult than MNIST because of its rich color, texture and shape.

Experimental results show that although other cognitive models (e.g., S1C1-SNN and CSNN) are hierarchical structures, the performance is still limited to the shallow frameworks and the encoding rule. The proposed joint CNN-SNN network can perform better, the main reasons may lie in its the deep structure and NFTI encoding rules which is more suitable than temporal encoding in deep structures. Compared with FTI, NFTI method utilized non fixed time interval which is more naturally in biological neuroscience, because we cannot fixed the time interval between 2 firing spikes in advance. Employing with the reasonable artitecture and NFTI encoding rules, the joint CNN-SNN is better to

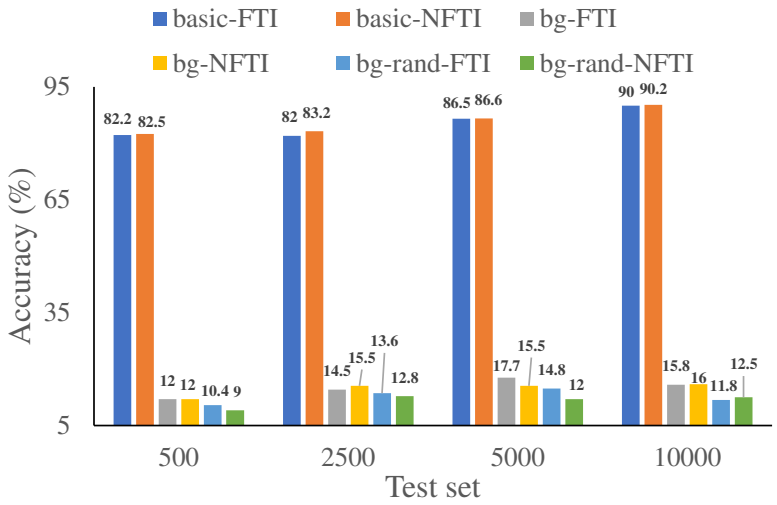



Fig. 4: Classification accuracy (%) of joint CNN-SNN model on noisy MNIST within FTI and NFTI spike firing rate encoding methods.

TABLE I: Evaluation on noisy MNIST and CIFAR10 from S1C1-SNN/CSNN/**CNN-SNN**.

| Dataset | MNIST | | | CIFAR10 | | | |
|---|---|---|---|---|---|---|---|
| Training | basic | basic | basic | $\sigma = 0$ | $\sigma = 0.01$ | $\sigma = 0.05$ | $\sigma = 0.1$ |
| Test | basic | bg | bg-rand | $\sigma = 0$ | $\sigma = 0.01$ | $\sigma = 0.05$ | $\sigma = 0.1$ |
| 500 | 76.0/81.0/**82.5** | 30.0/27.0/12.0 | 19.0/38.0/9.0 | 52.0/54.0/**58.0** | 50.0/51.0/51.4 | 45.0/45.0/48.0 | 32.0/33.0/33.5 |
| 1000 | 78.5/87.0/**86.5** | 29.0/26.0/12.5 | 14.5/23.5/11.5 | 56.0/55.0/**60.4** | 50.4/52.5/53.0 | 46.8/48.5/49.0 | 33.4/33.6/37.6 |
| 5000 | 77.3/84.7/**86.6** | 32.4/23.5/15.5 | 15.6/25.9/12.0 | 62.5/62.2/**63.7** | 51.2/54.0/58.3 | 48.0/50.6/52.8 | 34.9/35.8/39.5 |
| 10000 | 77.4/86.1/**90.2** | 31.3/22.3/16.0 | 12.3/24.0/12.5 | 68.2/68.6/**70.3** | 60.8/60.5/62.2 | 48.8/49.8/51.9 | 34.4/33.3/45.8 |
| 40000 | 76.0/83.8/**91.4** | 29.2/34.7/24.0 | 14.3/24.4/20.2 | 69.8/71.2/**73.5** | 60.6/61.2/63.5 | 49.0/50.6/52.8 | 35.5/35.2/50.3 |

extract important features than S1C1-SNN and CSNN systems.

### C. Evaluation of Parameter Quantization

One of core problems in implementing efficient spiking based models is that the parameters of SNNs are in floating point real value format. Dot product between real values leads to a huge power consumption which also brings the inconvenience to fixed-pointed communication based hardware implementation. To further save the energy of the proposed CNN-SNN system, we design a novel parameter quantization method which is very suitable to neuromorphic hardware platforms.

Since floating-point arithmetic takes too much hardware to implement, this paper implemented fixed-point arithmetic in SNN part of the proposed CNN-SNN model.

We determine the scaling factors $\beta$ and $\gamma$ as the upper limits of the cores parameters such as membrane potential $V_{max}$ and weight $W_{max}$. Firstly, we find the largest weight magnitude, then run the floating-point model with typical input, and find the largest magnitude of membrane potential. Then we compute the raw scaling factors by dividing the largest value of the fixed-point representation by the largest magnitudes. Finally, we round the raw scaling factors to the nearest power of 2.

The detailed processing of scaling factor searching from floating-point model is as following:

1) Find $W_{max} = MAX(|W|)$
2) Run the floating-point model with typical input, and find $V_{max} = MAX(|V_m|)$
3) Determine the intermediate scale factor $\beta_1 = \frac{2^{V_{bit}-1}}{V_{max}}$ and $\gamma = \frac{2^{W_{bit}-1}}{w_{max}}$ you can define the $V_{bit}$ and $W_{bit}$ which depend on the hardware limitations.
4) Make a log transform to determine the final scale factor $\beta = 2^{\lfloor \log_2 \beta_1 \rfloor}$ and $\gamma = 2^{\lfloor log_2 \gamma_1 \rfloor}$.

Based on the two scale factors $\beta$ and $\gamma$, we can implement the quantization process as the algorithm 1 described. If the spiking neuron was not in refractory period, we can scale the membrane potential $V$ and synapse weight $W$ respectively, until the scaled corresponding membrane potential exceed the threshold $V_{th}$, and the neuron fired a spike. Then let the neuron's membrane potential reset to 0, the neuron entered the refractory period. When the refractory period expired, the neuron moved to the next step of simulation. During the information communication in algorithm 1, all of the formats weights of the synapses, membrane potential of neurons and the spike signals are fixed-pointed which could reduce the memory and accelerate the calculation process compared to floating-point value, especially when the proposed model was implemented on neuromorphic hardware chips.

We adopt different bit-width quantization methods to show the efficiency of the proposed model. Compared to amount of synapses (W), the amount of neuron membrane potential (V) is fewer. Assuming a 400 $\times$ 400 fully-connected spiking neural network model which has 800 neuron membrane potential and 160,000 synapses, so the amount of synapses take up most of

**Algorithm 1** Quantization method for joint CNN-SNN

**Require:** Trained CNN-SNN model
**Ensure:** Quantized CNN-SNN model
After getting features from CNN output from CNN-SNN;
**if** The SNN part was not in refractory period **then**
  Let $(\beta V_m(t)) = ((\beta V_m(t-1)) \cdot V_{dec})$
  Then $(\beta V_m(t)) = (\beta V_m(t)) + (\frac{\beta}{\gamma}) \cdot \sum_j (\gamma W_j)$ (if the $j_{th}$ synapse generate a spike);
  **if** $\beta V_m(t) > \beta V_{th}$ **then**
    Output a spike;
    Let $\beta V_m(t) = 0$;
    Enter the refractory period;
  **end if**
**else**
  Exit refractory period when the refractory period expires;
**end if**

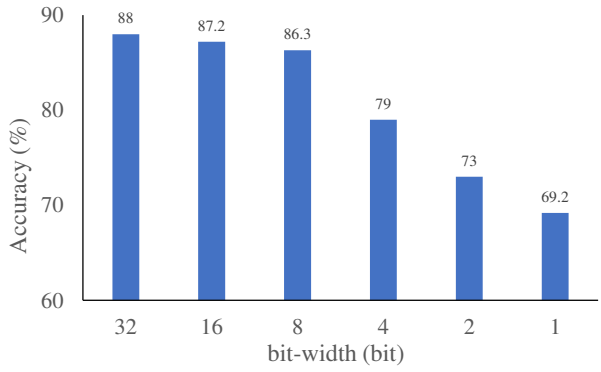



Fig. 5: Accuracy and memory comparison of different bit-width quantization.

memory. Thus, if we fixed more floating-point synapses as fixed-point, we can reduce lots of memory and accelerate the inference process in theory.

We adopted the basic MNIST as the experimental data, and the joint CNN-SNN model was trained on 2000 pictures and tested on 1000 pictures. To further exploit the computation efficiency of spiking based models, we set five different bit-width (fixed-point $V$ and $W$) to reduce the memory overhead in SNN part of the proposed model. Because the training samples are less than 10000, we set the both number of excitatory and inhibitory neurons to 100, thus, the quantized parameters are 10100 (10000 synapses and 100 neurons membrane potential). The evaluation results was shown in figure 5, we can observe that if we choose the 32 bit fixed-point as the quantized parameter format, the classification rate is 88% which is the same as the original floating-point 32 bit parameters, and it took up full memory as 32 bit floating-point value. When the parameters were quantized with decreasing as the power of two such as 16 bit, 8 bit until 1 bit, the classification accuracy of the proposed model is decreasing, but the memory usage dropped dramatically. When the bit-width of parameters was limited as 1, the performance only get 69.2% which means the spiking neural model needs enough model complexity to perform rich neural dynamics.

Obviously, it is a trade-off between memory usage and accuracy in the proposed model, and the fixed-point value is more friendly to neuromorphic chips compared with floating-point value.

### *D. Performance Comparison*

The proposed spiking based model achieves good classification on the MNIST, CIFAR10, and their noisy versions with the combination of CNN and the SNN. In Table II, we further illustrate the performance comparison with some of the most advanced SNN based brain-inspired frameworks on benchmark basic MNIST for seeing an overall picture.

Since we do not fix the size of SNN, the scale of each model can be varied. Based on the network capacity, the number of training and test samples would be adapted. From Table II, we can observe the related information and test accuracies of different spiking systems on the clean MNIST. It can be found that when we limit the number of training samples (i.e., with 500 training samples), the joint CNN-SNN achieves an accuracy rate of 82.5%, which is 4.5% higher than that S1C1-SNN with the same experimental settings.

With regard to the CSNN and Multi-Net, they are both embedded in supervised learning rules and Temporal encoding methods. The network capacity (only 300 neurons) limits the performance of CSNN, the accuracy is 87.0% on 10000 training samples. The best performance is achieved by the Multi-Net with 91.6% classification rate on only 2000 training-sample datasets. Spiking RBM, Dendritic Neurons, and Un-stdp could get 89.0%, 90.3%, and 90.6% classification accuracies with different numbers of training samples. Promising performance is obtained with a large collection of training samples or supervised learning rules.

Despite the joint CNN-SNN cannot achieve the best performance (90.2%), it can attain comparable accuracy on small-size training sets. Especially when the bit-width of the proposed model was scaled to 16, the proposed model still has 87.2% classification performance, and the memory usage dropped 50% compared to 32 bit floating-point value.

Compared with the Multi-Net (91.6%), our model is lighter. The Multi-Net has more parameters (71,026 biological neurons) than joint CNN-SNN possesses (800 biological neurons), more numbers and bit-width of parameters means higher computational consumption.

## IV. Conclusion

In this work, a hierarchical feature extraction enhanced spiking model called joint CNN-SNN is presented. Combining the enhanced feature extraction with FTI/NFTI encoding mechanisms, this visual simulation framework is tailed to encode the external stimuli (images) into spatio-temporal patterns with rich neural dynamics. We demonstrate the proposed framework implemented to MNIST, CIFAR10, and the corresponding variations can obtain comparable performance with other spiking based systems: S1C1-SNN, CSNN, spiking RBM, Dendritic Neurons, Un-stdp and Multi-Net.

TABLE II: Accuracy comparison of spiking based models on clean MNIST dataset.

| Networks | Encoding Methods | Training Rules | Training/test samples | Perfromance (%) |
|---|---|---|---|---|
| S1C1-SNN [12] | Temporal | Tempotron (supervised) | 500/100 | 78.0 |
| CSNN [11] | Temporal | Tempotron (supervised) | 10000/2000 | 87.0 |
| Spiking RBM [38] | Rate-based | Contrastive divergence (supervised) | 60000/10000 | 89.0 |
| Dendritic Neurons [39] | Rate-based | Morphology learning (supervised) | 10000/5000 | 90.3 |
| Un-stdp [37] | Rate-based | STDP (unsupervised) | 40000/8000 | 90.6 |
| Multi-Net [40] | Temporal | STDP with calcium (supervised) | 2000/1000 | 91.6 |
| **Joint CNN-SNN (this paper)** | **NFTI Rate-based** | **STDP (unsupervised)** | **500/100** | **82.5** |
| **Joint CNN-SNN (this paper with 16 bit fixed-point parameters)** | **NFTI Rate-based** | **STDP (unsupervised)** | **2000/1000** | **87.2** |
| **Joint CNN-SNN (this paper)** | **NFTI Rate-based** | **STDP (unsupervised)** | **2000/1000** | **88.0** |
| **Joint CNN-SNN (this paper)** | **NFTI Rate-based** | **STDP (unsupervised)** | **10000/2000** | **90.2** |

Experimental results indicate that the reasonable structure and encoding methods employed by joint CNN-SNN can benefit to extract more significant feature presentations, transferring them to spatiotemporal spike trains and obtaining a more neuromorphic oriented spiking model through a parameter quantized method.

Because the proposed model used a CNN as feature extraction, it would be more helpful to extract features from static images compared to time-series data such as output from dynamic vision sensors (DVS) [41]. Advancements in event based sensors align with the development of neuromorphic chips and devices, where the data format is events or spikes. In our next step, we would like to build a pure spike in-spike out model to handle event-based computing, besides the event based sensors, and implement the proposed model in our proposed neuromorphic hardware platform [42].

## REFERENCES


[1] S. Thorpe, D. Fize, and C. Marlot, "Speed of processing in the human visual system." *Nature*, vol. 381, no. 6582, pp. 520–522, 1996.

[2] J. J. DiCarlo, D. Zoccolan, and N. C. Rust, "How does the brain solve visual object recognition?" *Neuron*, vol. 73, no. 3, pp. 415–434, 2012.

[3] J. Yang, P. Zhang, and Y. Liu, "Robustness of classification ability of spiking neural networks," *Nonlinear Dynamics*, vol. 82, no. 1, pp. 723–730, 2015.

[4] C. P. Hung, G. Kreiman, T. Poggio, and J. J. Dicarlo, "Fast readout of object identity from macaque inferior temporal cortex," *Science*, vol. 310, no. 5749, pp. 863–866, 2005.

[5] W. Fang, Z. Yu, Y. Chen, T. Masquelier, T. Huang, and Y. Tian, "Incorporating learnable membrane time constant to enhance learning of spiking neural networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2661–2671.

[6] J. Hu, H. Tang, K. C. Tan, and H. Li, "How the Brain Formulates Memory: A Spatio-Temporal Model Research Frontier," *IEEE Computational Intelligence Magazine*, vol. 11, no. 2, pp. 56–68, 2016.

[7] M. Mattia, M. Biggio, A. Galluzzi, and M. Storace, "Dimensional reduction in networks of non-markovian spiking neurons: Equivalence of synaptic filtering and heterogeneous propagation delays," *PLoS computational biology*, vol. 15, no. 10, p. e1007404, 2019.

[8] B. A. y Arcas and A. L. Fairhall, "What causes a neuron to spike?" *Neural Computation*, vol. 15, no. 8, pp. 1789–1807, 2003.

[9] M. Zhang, J. Wang, J. Wu, A. Belatreche, B. Amornpaisannon, Z. Zhang, V. P. K. Miriyala, H. Qu, Y. Chua, T. E. Carlson *et al.*, "Rectified linear postsynaptic potential function for backpropagation in deep spiking neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 5, pp. 1947–1958, 2021.

[10] Q. Xu, J. Shen, X. Ran, H. Tang, G. Pan, and J. K. Liu, "Robust transcoding sensory information with neural spikes," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 5, pp. 1935–1946, 2021.

[11] Q. Xu, Y. Qi, H. Yu, J. Shen, h. Tang, and G. Pan, "CSNN: An Augmented Spiking based Framework with Perceptron-Inception," in *Twenty-Seventh International Joint Conference on Artificial Intelligence*, 2018, pp. 3560–3566.

[12] Q. Yu, H. Tang, K. C. Tan, and H. Li, "Rapid feedforward computation by temporal encoding and learning with spiking neurons." *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 10, pp. 1539–1552, 2013.

[13] S. M. Bohte, J. N. Kok, and H. La Poutre, "Error-backpropagation in temporally encoded networks of spiking neurons," *Neurocomputing*, vol. 48, no. 1-4, pp. 17–37, 2002.

[14] B. Petro, N. Kasabov, and R. M. Kiss, "Selection and optimization of temporal spike encoding methods for spiking neural networks," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 2, pp. 358–370, 2019.

[15] M. Nelson and J. Rinzel, "The hodgkin—huxley model," in *The book of genesis*. Springer, 1998, pp. 29–49.

[16] W. Gerstner, "A framework for spiking neuron models: The spike response model," in *Handbook of Biological Physics*. Elsevier, 2001, vol. 4, pp. 469–516.

[17] W. H. Calvin and C. F. Stevens, "Synaptic noise and other sources of randomness in motoneuron interspike intervals." *Journal of neurophysiology*, vol. 31, no. 4, pp. 574–587, 1968.

[18] J. Wu, C. Xu, X. Han, D. Zhou, M. Zhang, H. Li, and K. C. Tan, "Progressive tandem learning for pattern recognition with deep spiking neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[19] T. Gollisch and M. Meister, "Rapid neural coding in the retina with relative spike latencies," *Science*, vol. 319, no. 5866, pp. 1108–11, 2008.

[20] S. R. Kheradpisheh, M. Ganjtabesh, S. J. Thorpe, and T. Masquelier, "Stdp-based spiking deep convolutional neural networks for object recognition," *Neural Networks*, vol. 99, pp. 56–67, 2018.

[21] Q. Yu, S. Song, C. Ma, J. Wei, S. Chen, and K. C. Tan, "Temporal encoding and multispike learning framework for efficient recognition of visual patterns," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[22] G. Orchard, C. Meyer, R. Etienne-Cummings, C. Posch, N. Thakor, and R. Benosman, "HFirst: A Temporal Approach to Object Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 10, pp. 2028–2040, 2015.

[23] X. Cheng, Y. Hao, J. Xu, and B. Xu, "Lisnn: Improving spiking neural networks with lateral interactions for robust object recognition." in *IJCAI*, 2020, pp. 1519–1525.

[24] X. Lagorce, G. Orchard, F. Galluppi, B. E. Shi, and R. B. Benosman, "HOTS: A Hierarchy of Event-Based Time-Surfaces for Pattern Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, pp. 1–1, 2017.

[25] M. Uzuntarla, M. Ozer, U. İleri, A. Calim, and J. J. Torres, "Effects of dynamic synapses on noise-delayed response latency of a single neuron," *Physical Review E*, vol. 92, no. 6, p. 062710, 2015.

[26] S. R Angus, "Neuronal arithmetic," *Nature Reviews Neuroscience*, vol. 11, no. 7, pp. 474–89, 2010.

[27] A. J. Yu, M. A. Giese, and T. Poggio, "Biophysiologically Plausible Implementations of the Maximum Operation," *Neural Computation*, vol. 14, no. 12, p. 2857, 2002.

[28] O. Peter, N. Daniel, S. C. Liu, D. Tobi, and P. Michael, "Real-time classification and sensor fusion with a spiking deep belief network," *Frontiers in Neuroscience*, vol. 7, p. 178, 2013.

[29] J. Wu, Y. Chua, M. Zhang, Q. Yang, G. Li, and H. Li, "Deep spiking neural network with spike count based learning rule," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–6.

[30] X. A. Qi, B. Jp, C. Jsa, A. Ht, and P. Gang, "Deep covdensesnn: A hierarchical event-driven dynamic framework with spiking neurons in noisy environment," *Neural Networks*, vol. 121, pp. 512–519, 2020.

[31] J. Hu, H. Tang, K. C. Tan, H. Li, and L. Shi, "A spike-timing-based integrated model for pattern recognition." *Neural Computation*, vol. 25, no. 2, pp. 450–472, 2013.

[32] Y. Lécun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[33] H. Larochelle, D. Erhan, A. Courville, J. Bergstra, and Y. Bengio, "An empirical evaluation of deep architectures on problems with many factors of variation," in *International Conference on Machine Learning*, 2007, pp. 473–480.

[34] J. Shen, Y. Zhao, J. K. Liu, and Y. Wang, "Hybridsnn: Combining bio-machine strengths by boosting adaptive spiking neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[35] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[36] G. Dan and R. Brette, "Brian: A Simulator for Spiking Neural Networks in Python." *Bmc Neuroscience*, vol. 9, no. 1, pp. 1–2, 2008.

[37] P. U. Diehl and M. Cook, "Unsupervised learning of digit recognition using spike-timing-dependent plasticity," *Frontiers in Computational Neuroscience*, vol. 9, p. 99, 2015.

[38] P. Merolla, J. Arthur, F. Akopyan, N. Imam, R. Manohar, and D. S. Modha, "A digital neurosynaptic core using embedded crossbar memory with 45pJ per spike in 45nm," in *Custom Integrated Circuits Conference*, 2011, pp. 1–4.

[39] S. Hussain, S. C. Liu, and A. Basu, "Improved margin multi-class classification using dendritic neurons with morphological learning," in *IEEE International Symposium on Circuits and Systems*, 2014, pp. 2640–2643.

[40] M. Beyeler, N. D. Dutt, and J. L. Krichmar, "Categorization and decision-making in a neurobiologically plausible spiking network using a STDP-like learning rule," *Neural Netw*, vol. 48, no. 10, pp. 109–124, 2013.

[41] O. Bichler, D. Querlioz, S. J. Thorpe, J.-P. Bourgoin, and C. Gamrat, "Extraction of temporally correlated features from dynamic vision sensors with spike-timing-dependent plasticity," *Neural networks*, vol. 32, pp. 339–348, 2012.

[42] D. Ma, J. Shen, Z. Gu, M. Zhang, X. Zhu, X. Xu, Q. Xu, Y. Shen, and G. Pan, "Darwin: A neuromorphic hardware co-processor based on spiking neural networks," *Journal of Systems Architecture*, vol. 77, pp. 43–51, 2017.


**Qi Xu** received his B.Eng. degree in the College of Computer Science and Technology from Zhejiang University of Technology in 2015, and Ph.D. degree in the College of Computer Science and Technology from Zhejiang University in 2021. Since 2021, he has been a tenure-track associate professor at School of Artificial Intelligence, Dalian University. He was ever granted as an honorary visiting fellow in the Centre for Systems Neuroscience, University of Leicester, U.K in 2019. His research interests include brain-inspired computing, neuromorphic computing, neural computation, computational neuroscience, and cyborg intelligence.

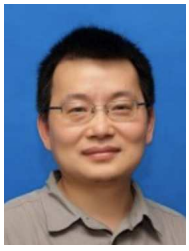
**Yaxin Li** received her Bachelor's degree in the School of Computer Science and Information Engineering from Hefei University of Technology in 2022, and working on her master's degree at School of Artificial Intelligence, Dalian University of Technology. Her research interests include neuromorphic computing, spiking neural network, neural computation and computational neuroscience.

**Jiangrong Shen** received her Bachelor's degree in the Department of Computer Science and Technology from Hebei University in 2015, and Ph.D. degree in the College of Computer Science and Technology from Zhejiang University in 2021. Currently, she is a postdoctoral fellow at the College of Computer Science and Technology, Zhejiang University. She studied as an honorary visiting scholar in the Department of Neuroscience, Psychology and Behaviour, University of Leicester in 2019. Her research interests include neuromorphic computing, neural computation, and cyborg intelligence.

**Pingping Zhang** received the BE degree in mathematics and applied mathematics from Henan Normal University, Xinxiang, China, in 2012 and the PhD degree in signal and information processing from the Dalian University of Technology (DUT), Dalian, China, in 2020. He is currently an associate professor with the School of Artificial Intelligence, DUT. His research interests include deep learning, saliency detection, object tracking, and semantic segmentation.

**Jian K. Liu** Jian K. Liu received the Ph.D. degree in mathematics from the University of California at Los Angeles, Los Angeles, CA, USA, in 2009. He is currently a Lecturer with School of Computing, University of Leeds, Leeds, U.K. His current research interests include computational neuroscience and brain-like computation.

**Huajin Tang** (Member, IEEE) received the Ph.D.degree from the National University of Singapore, Singapore, in 2005. He was a Research and Development Engineer with STMicroelectronics, Singapore, from 2004 to 2006. From 2006 to 2008, he was a Post-Doctoral Fellow with the Queensland Brain Institute, the University of Queensland, Saint Lucia, QLD, Australia. He was the Head of the Robotic Cognition Laboratory, Institute for Infocomm Research, Singapore, from 2008 to 2015. He is currently a Professor with Zhejiang University. His research interests include neuromorphic computing, neuromorphic hardware and cognitive systems, robotic cognition.



**Gang Pan** (Member, IEEE) received the B.Eng. and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 1998 and 2004, respectively. From 2007 to 2008, he was a Visiting Scholar with the University of California at Los Angeles, Los Angeles, CA, USA. He is currently a Professor with the Department of Computer Science and the Deputy Director of the State Key Laboratory of CAD&CG, Zhejiang University. He has authored over 100 refereed articles and 35 patents granted. His current interests include artificial intelligence, pervasive computing, brain-inspired computing, and brain-machine interfaces.