Hindawi
Neural Plasticity
Volume 2020, Article ID 8851351, 11 pages
https://doi.org/10.1155/2020/8851351

Research Article

Pattern Recognition of Spiking Neural Networks Based on Visual Mechanism and Supervised Synaptic Learning

Xiumin Li, Hao Yi, and Shengyuan Luo

College of Automation, Chongqing University, Chongqing 400044, China

Correspondence should be addressed to Xiumin Li; free_xmin@hotmail.com

Received 5 August 2020; Revised 9 October 2020; Accepted 14 October 2020; Published 28 October 2020

Academic Editor: Rubin Wang

Electrophysiological studies have shown that mammalian primary visual cortex are selective for the orientations of visual stimuli. Inspired by this mechanism, we propose a hierarchical spiking neural network (SNN) for image classification. Grayscale input images are fed through a feed-forward network consisting of orientation-selective neurons, which then projected to a layer of downstream classifier neurons through the spiking-based supervised tempotron learning rule. Based on the orientation-selective mechanism of the visual cortex and tempotron learning rule, the network can effectively classify images of the extensively studied MNIST database of handwritten digits, which achieves 96% classification accuracy based on only 2000 training samples (traditional training set is 60000). Compared with other classification methods, our model not only guarantees the biological plausibility and the accuracy of image classification but also significantly reduces the needed training samples. Considering the fact that the most commonly used deep learning neural networks need big data samples and high power consumption in image recognition, this brain-inspired computational neural network model based on the layer-by-layer hierarchical image processing mechanism of the visual cortex may provide a basis for the wide application of spiking neural networks in the field of intelligent computing.

1. Introduction

Pattern recognition is to distinguish different kinds of patterns according to the semantic information of inputs, and it is an important basic problem in the field of computer vision and artificial intelligence. There is a wide range of applications in reality, such as object recognition [1, 2], region detection [3], aurora image analysis [4], and scene categorization [5]. Due to the great research value, researchers have developed many methods for pattern recognition, for example, k-nearest neighbor classifier [6], multilayer perceptron [7], support vector machine [8], and convolutional neural networks (CNNs) [9, 10]. In these methods, CNNs are one of the most commonly used methods which has achieved great success in intelligent computing. Although CNNs implement a biological-inspired network topology, the node models and learning rules are quite different from the real-life biological neural networks. And CNNs often rely on the error backpropagation learning rule, which has been criticized for being biologically unrealistic and requiring massive computing power, time, and energy, and thus make it infeasible for real-time applications, e.g., on mobile devices or autonomous robots [11].

In fact, the learning method based on the error gradient descent lacks biological support. It is known that the human being has an overwhelming advantage in image recognition, which shows that our brain has a unique processing mechanism in visual processing. Therefore, many researchers try to find more biosupported classification methods by imitating the structure and synaptic plasticity learning rules of biological neural networks. Among various temporal learning rules for spiking neural models, several rules have been widely studied, including spike-timing-dependent plasticity (STDP), the tempotron rule, the SpikeProp rule, the SPAN rule, the Chronotron rule, and the ReSuMe rule [12]. Markram proposed the spike-timing-dependent plasticity (STDP) which has been confirmed to explain the brain activities especially for LTP and LTD [13]. STDP learning rule which adjusts the synaptic weights according to the spiking time interval of presynaptic and postsynaptic currents is widely used for unsupervised learning of SNNs [14, 15]. In view of the great success of deep learning, Kheradpisheh proposed a spiking deep convolutional neural network (SDNN) based on STDP rules, which can extract image features with multilayer convolution operation [16], Kulkarni proposed normalized approximate descent to apply error back propagation [17], and they both achieved high performance on the MNIST dataset. In other applications, Seijoon proposed spike Yolo, which is the first time to use the SNN for target detection and has achieved the same performance as CNN [18]. In particular, the tempotron learning is specially designed to solve the binary classification problem of spike trains. Compared with other algorithms, it has stronger pattern recognition ability and can make appropriate decisions based on an error-correcting strategy which adjusts only a few parameters under the effect of supervisory signals. Moreover, different from traditional supervised learning algorithm based on gradient descent, the tempotron learning does not need to explicitly calculate the gradient of the cost function with respect to parameters, which avoids the problem caused by the discontinuity of firings in SNNs.

Besides learning rules, preprocessing of input data and visual attention is also crucial for visual image recognition [19]. In 1959, Hubel and Wiesel inserted a electrode into the cat's visual cortex and projected light and dark pattern on the screen in front of the cat, and they found that some neurons in the visual cortex fired rapidly when the line was at a particular angle, while other angles corresponds to different neurons [20]. This experiment showed that neurons in the visual cortex are selectively responsive to the directions of visual stimulation, and such cells are known as orientation selective cells. Subsequently, Simoncelli established a mathematical model, called the motion energy model [21], for the response of visual cortical areas V1 and MT. And the MT receptive field is sensitive to pattern direction. Recently, Beyeler et al. modify the motion energy model and runs on off-the-shelf graphics processing unit (GPU) successfully [22]. Their simulator CARLsim is written in C/C++ and running on CUDA environment which is an efficient computing platform of NVIDIA, supporting many kinds of spiking neuron models and synaptic plasticity rules [23]. They built a hierarchical feedforward SNN that integrates a low-level memory encoding mechanism with a higher level decision process to perform a visual classification task in real-time. The network was trained by an STDP-like learning rule and finally achieved 92% accuracy in the MNIST database of handwritten digits [24].

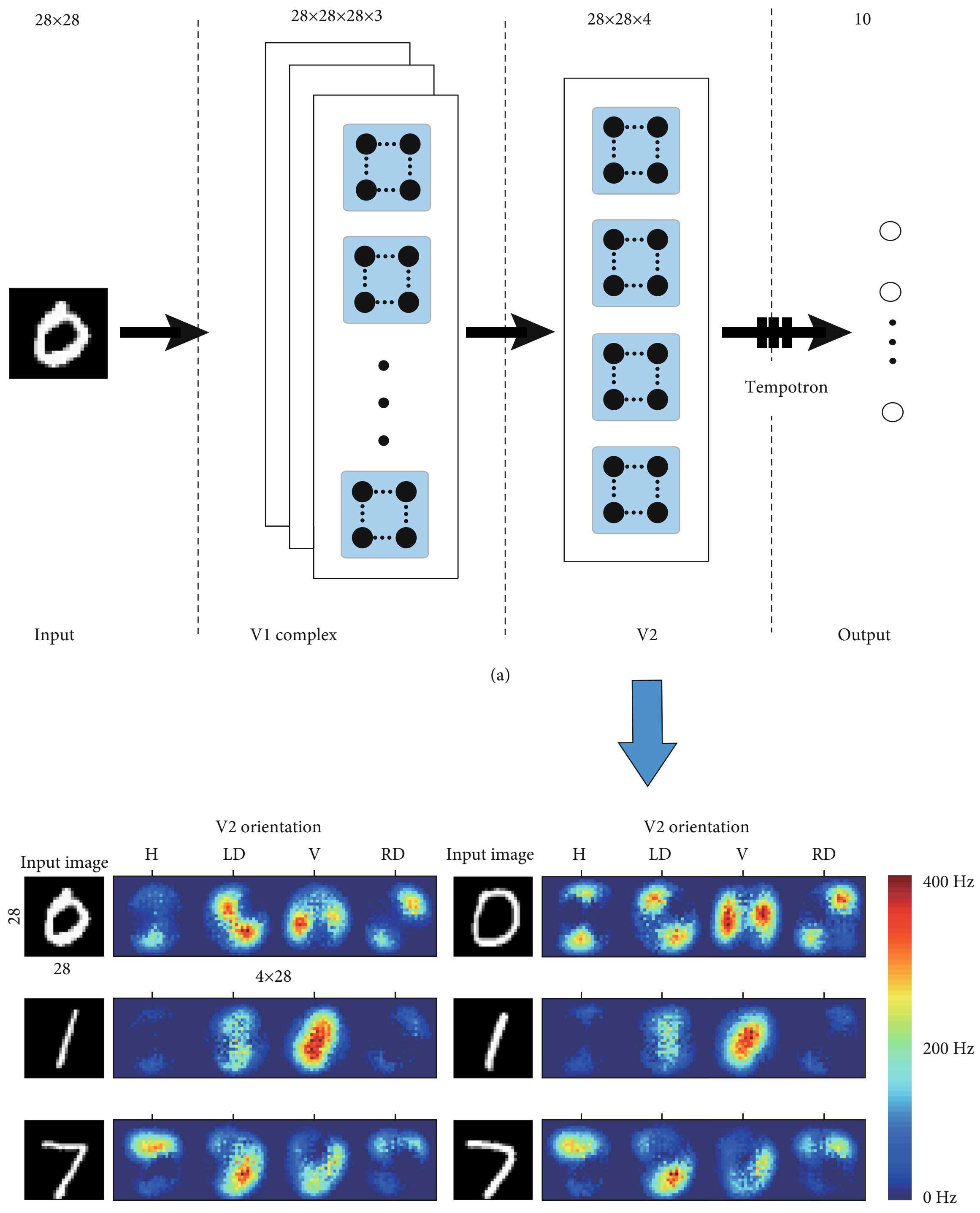
In general, although SNNs have strong biological characteristics and have been broadly studied in computational neuroscience, the computing accuracy is lower than that of traditional artificial neural networks. In this article, by applying the motion energy model and orientation-selective mechanism of visual cortex as studied in [24] for the preprocessing of input data, we use the biologically plausible supervised synaptic learning rule named tempotron [25] to replace the STDP-like learning rule to train the output weights. In order to simplify the training process and improve computational efficiency, we propose the single-spike temporal encoding which is one of the simplest temporal encoding for the transform of firing rates to spiking trains before the tempotron learning. Our results show that the this model is very efficient for the MNIST hand-written digit recognition with the recognition accuracy of 96% based on only 2000 training samples. Compared with other image classification methods, our model needs much smaller data samples to achieve high performance of classification.

2. Methods

2.1. Network Architecture. The network architecture is shown in Figure 1(a). The network is divided into four layers (dotted boxes). The input layer (MNIST grayscale images with 28×28 pixels) was fed through a feed-forward network consisting of V1 and V2, which then projected to a downstream population of decision neurons (output layer). V1 and V2 populations were previously implemented and tested in papers [21–23]. Each image is spatially expanded into four pools orientation-selective responses of the V2 layer neurons, with each pool of neurons preferentially responded to one of four spatial orientations in $45 \circ$ increments (horizontal "H," right diagonal "RD," vertical "V," and left diagonal "LD" (Figure 1(b)). V1 simple and complex cell responses were constructed by using the rate-based motion energy model. The 28×28 pixel images in the MNIST dataset were processed at three different spatial scales resulting in 84 filter responses per pixel. Simulated spiking neurons in V2 received input only from V1 complex cells and preferentially responded to one of four spatial orientations. The orientation layer V2 thus consists of four pools of 28×28 neurons (as illustrated in Figure 1(b)). Synaptic weights from V2 to the output layer are trained using the spiking-based tempotron supervised learning rule which will be introduced in the following sections. Figure 1(b) shows some MNIST samples (28×28) and their corresponding orientation-selective responses (28×112). The orientation layer consisted of four pools of 28×28 neurons responding to one of four spatial orientations. Although there are subtle differences in writing for the same digit, the heatmaps are very similar and significantly different from the other digits. For example, digit 1 shows the strongest response to the vertical direction. Firing rates are color-coded ranging from 0 Hz (blue) to 400 Hz (red).

3. Motion Energy Model

V1 consists of the V1 simple cell, V1 complex cell, and spiking neurons. V1 cell responses were constructed by using the first stage of the (rate-based) motion energy model [23]. V1 simple cells are modeled as linear space-time-oriented filter whose receptive fields, consisted of 28 space-time orientations (Uniform distribution in sphere), are third derivatives of a Gaussian which are very similar to a Gabor filter. V1 complex cell responses were computed as a weighted sum of simple cell afferents that have the same space-time orientation and phase. Since the motion energy model expected movies rather than static images as an input, we expand the 28×28 pixel image into videos (20 frames with each duration of 50 ms) as the input. A visual stimulus was processed at



(a)

(b)

Figure 1: (a) Network architecture. The network is divided into four layers (dotted boxes). The input layer (MNIST grayscale images with 28 × 28 pixels) was fed through a feed-forward network consisting of V1 and V2, which then projected to a downstream population of decision neurons (output layer). The black dot is the Izhikevich model, and the white circle is the leaky integrate-and-fire (LIF) model. The input is the raw pixel value of MINIST grayscale images. V1 simple and complex cell responses were constructed by using the rate-based motion energy model. Simulated spiking neurons in V2 received input only from V1 complex cells and preferentially responded to one of four spatial orientations. There are full connections from the V2 layer to the output layer, with the weights updated by the tempotron learning rule. (b) Some MNIST samples (28 × 28) and their corresponding orientation-selective responses (28 × 112). Each image is spatially expanded into four pool orientation-selective responses of the V2 layer neurons, with each pool of neurons preferentially responded to one of four spatial orientations (horizontal “H,” right diagonal “RD,” vertical “V,” and left diagonal “LD”). Firing rates are color-coded ranging from 0Hz (blue) to 400Hz (red).

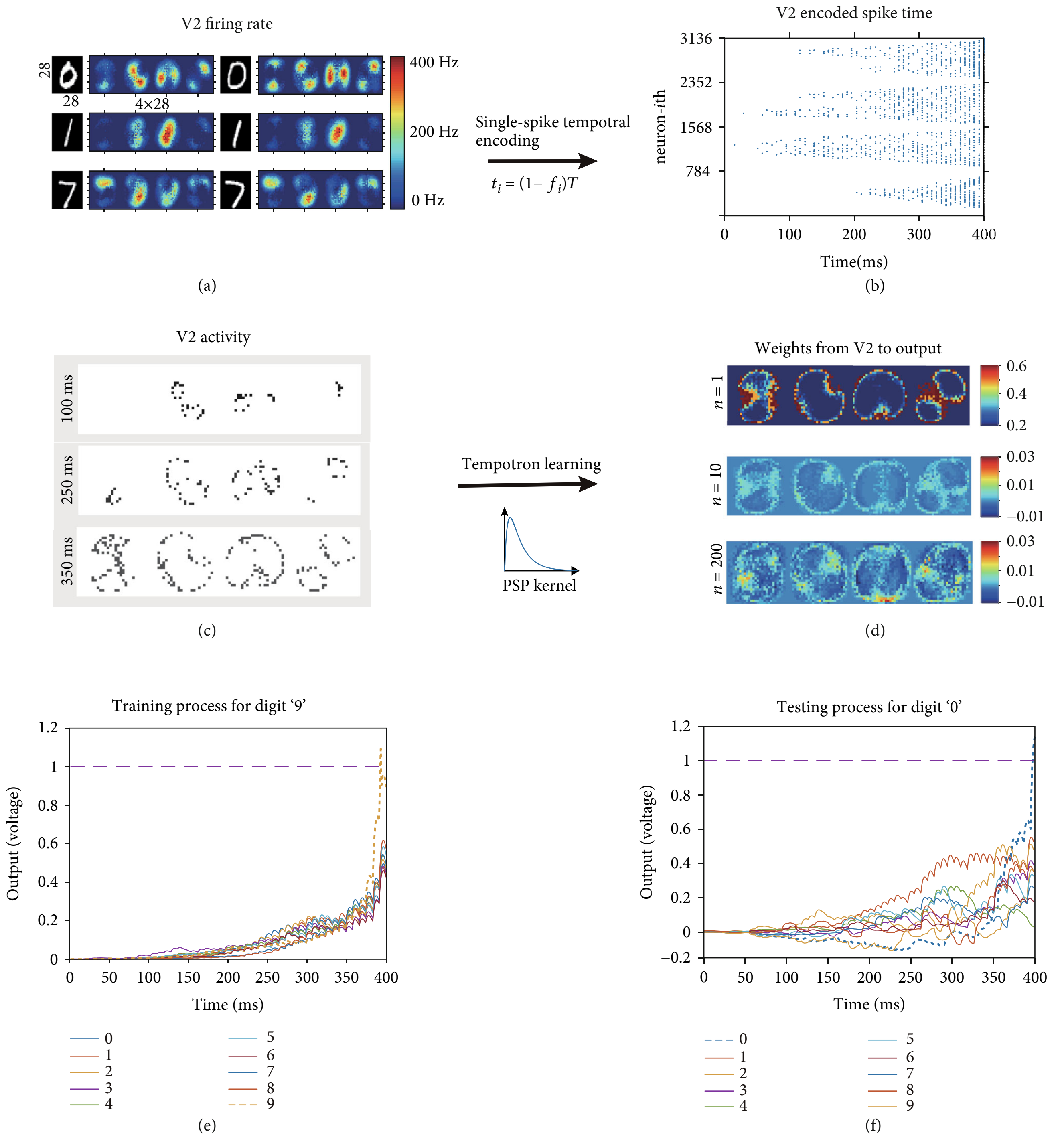


Figure 2: (a) Firing rates of four pool orientation-selective V2 neurons for different digit images. (b) Spike trains for V2 neurons after $t =$ 100,250,350ms. (d) Synaptic weights from the V2 layer to the output layer after tempotron learning with $n = 1,10,200$ training samples for the digit "0." Note that firing activities of the edge neurons of each pool of the orientation population in V2 as shown in (c) (bottom panel) are quite similar with the weight matrix after learning. (e, f) The voltage accumulations of output neurons during the training and testing process, respectively.

three different spatiotemporal scales (as the three banks depicted in neuron populations in the V1 layer of Figure 1) resulting in 84 filter responses per pixel. Filter responses were then interpreted as the mean firing rates of Poisson spike trains which are the input of V1 spiking neurons.

V2 only consists of spiking neurons which receive the input from V1 spiking neurons and more sensitive to four specified directions in 45° increments (horizontal "H," right-diagonal "RD," vertical "V," and left-diagonal "LD"). The neurons were broadly tuned such that they only response strongest to a specific direction in a Gaussian neighborhood and send an inhibitory projection to the neurons with the antipreferred direction. V2 orientation layer consisted of 3136 neurons correspond to 28×28 pixels in four orientations. The orientation information is presented by the firing rate of the V2 neuron population. For complete model description and parameter settings, please refer to [21–23]. Here, V1 and V2 layers are modeled by using the CARLSim SNN simulator platform which is available on the website http://www.socsci.uci.edu/jkrichma/CARLsim/.

4. Spiking Neuron Model

The neurons in V1 and V2 are modeled by using the Izhikevich neuron [26], which has both computational efficiency and biological plausibility for simulating large-scale spiking neural networks.

$$\begin{gathered} dV(t)/dt = 0.04V(t)^2 + 5V(t) + 140 - U(t) + I^{\mathrm{syn}}(t) \\ dU(t)/dt = a(bV(t) - U(t)), \\ \text{if } V > 30\mathrm{mV}, \text{ then} \begin{cases} V = c \\ U = U + d \end{cases}, \end{gathered} \tag{1}$$

where V represents the membrane potential, and U is a membrane recovery variable; I^{syn} is the external synaptic current through neuron, and the units for the membrane potential V and time t are mv and ms, respectively. The parameters a, b, c, d can be set as different values according to different neuron types. For example, regular spiking (RS) neurons (excitatory neurons) have $a = 0.02, b = 0.2, c = -65, d = 8$, and fast spiking (FS) neurons (inhibitory neurons) have $a = 0.1, b = 0.2, c = -65, d = 2$ [26, 27].

There are two synapse model descriptions: the current-based (CUBA) description uses a single synaptic current term while the conductance-based (COBA) description calculates the synaptic current using more complex conductance equations for each synaptic receptor-type. Both CUBA and COBA current contributions are influenced by the synaptic weight of the synapse [28]. Here, we use the COBA mode. There are four receptor types, that is, AMPA (fast decay), NMDA (slow decay and voltage-dependent), GABA$_a$ (fast decay), or GABA$_b$ (slow decay). A spike arriving at a synapse that is postsynaptically connected to an excitatory (inhibitory) neuron increases both g_{AMPA} and g_{NMDA} (g_{GABA_a} and g_{GABA_b}). So, I_{syn} includes the excitatory current and inhibitory current:

$$\begin{aligned} I_{\mathrm{syn}} &= i_e + i_i \\ i_e &= i_{\mathrm{AMPA}} + i_{\mathrm{NMDA}} . \\ i_i &= i_{\mathrm{GABA}_a} + i_{\mathrm{GABA}_b} \end{aligned} \tag{2}$$

The total current I_{syn} for each neuron is given by

$$\begin{aligned} i_{\mathrm{AMPA}} &= g_{\mathrm{AMPA}}\left(v - v_{\mathrm{AMPA}}^{\mathrm{rev}}\right) \\ i_{\mathrm{NMDA}} &= g_{\mathrm{NMDA}} \frac{[v + 80/60]^2}{1 + [v + 80/60]^2}\left(v - v_{\mathrm{NMDA}}^{\mathrm{rev}}\right) \quad , \\ i_{\mathrm{GABA}_a} &= g_{\mathrm{GABA}_a}\left(v - v_{\mathrm{GABA}_a}^{\mathrm{rev}}\right) \\ i_{\mathrm{GABA}_b} &= g_{\mathrm{GABA}_b}\left(v - v_{\mathrm{GABA}_b}^{\mathrm{rev}}\right) \end{aligned} \tag{3}$$

where g and v_{rev} are specific to a particular ion channel or receptor. Synaptic conductances g obey the exponential decay and changed when presynaptic spikes arrived.

$$\frac{dg_r(t)}{dt} = -\frac{1}{\tau_r} g_r(t) + w \sum_i \delta(t - t_i), \tag{4}$$

where δ is the Dirac delta function, w is the synaptic weight, τ is the time constant, and r is the receptor type (AMPA, NMDA, GABA$_a$, GABA$_b$). t_i is the arriving time of presynaptic spikes. In our simulations, we set the parameters as $\tau_{\mathrm{AMPA}} = 5\mathrm{ms}$, $\tau_{\mathrm{NMDA}} = 150\mathrm{ms}$, $\tau_{\mathrm{GABA}_a} = 6$ ms, and $\tau_{\mathrm{GABA}_b} = 150\mathrm{ms}$.

5. Single-Spike Temporal Encoding for V2

Since the orientation information in the V2 layer is presented by firing rates of the neuronal population (shown in Figure 1(b)), it cannot be directly transmitted to the output neurons for the spiking-based supervised learning. Here, we adopt the latency coding [29] for the transform of V2 firing rates to spiking trains. Latency coding is one of the simplest temporal encoding. It encodes information of the response time in the encoding window, which is related to external stimulus. According to the characteristics of neurons, i.e., the larger the input, the earlier the neuron discharges, and no firing without stimulation, the input information is encoded into a delay time relative to the initial time and mapped to a predefined time window. In order to simplify the training process and improve computational efficiency, we normalize the firing frequencies of V2 neurons and encode them into a spiking train in the time window between 0 and T with each neuron fires only once (i.e., single-spike temporal encoding). V2 neurons with high firing rates would result in earlier encoded spiking time, whereas neurons with low firing rates would firing later or no firing at all (as shown in Figures 2(a)–2(c)). In this way, the spike train of the encoded V2 layer is sparse and much efficient for tempotron learning. The spiking time of the ith V2 neuron is obtained by

$$t_i = (1 - f_i)T. \tag{5}$$

Here, f_i is the normalized firing frequency of the ith ($i = 1, 2, ,3136$) neuron in the V2 layer. The length of time window is $T = 400\mathrm{ms}$. After encoding, the higher the firing frequency, the earlier the spike time t_i. For example, if the i

th V2 neuron has no firings, its encoded spike time t_i is equal to T. Note that the spiking time T is not included in the calculations of synaptic learning.

6. Tempotron Supervised Learning

Following the previous implementation of V1 and V2 layers and the single-spike temporal encoding, we train the synaptic weights from V2 to the output layer for the recognition of the MNIST database using a spiking-based supervised learning rule, called the tempotron learning proposed in [25]. The V2 orientation layer is fully connected to the output layer which is plastic in Figure 1. The output layer consists of 10 leaky integrate-and-fire (LIF) neurons, corresponding to 10 types of digits $(0, 1, \cdots, 9)$. There are 3136 × 10 connections (presynaptic 3136 neurons times postsynaptic 10 output neurons), and the synaptic weights w are a matrix of 3136×10. The subthreshold membrane voltage of the decision/output layer is a weighted sum of the postsynaptic potentials (PSPs) from all incoming spikes of the V2 layer [25]:

$$V_j(t) = \sum_{i=1}^{N} w_{ij} \sum_{t_i} K(t - t_i) + V_{\text{rest}}, \tag{6}$$

where t_i is the spiking time of the ith neuron $(i = 1, 2, ,\cdots, N, N$ is the number of presynaptic V2 neurons), and w_{ij} is the synaptic weights from the ith V2 layer neuron to the jth output neuron. $K(t - t_i)$ is the normalized PSP contributed by each incoming spike:

$$K(t - t_i) = V_0 \left(\frac{\exp\left(-(t - t_i)\right)}{\tau} - \frac{\exp\left(-(t - t_i)\right)}{\tau_s} \right), t_i < t, \tag{7}$$

Here, τ and τ_s denote decay time constants of membrane integration and synaptic currents, respectively. The factor V_0 normalizes PSP kernels to 1. Input spike arrives in the time t which locates in the time window $[0, T]$. When $V(t)$ crossed the firing threshold (here, $V_{\text{threshold}} = 1$), the neuron elicits a spike, then $V(t)$ exponentially decays by shunting all incoming spikes that arrived after the output spike (for details, please see [25]). Here, we take $V_{\text{rest}} = 0$, $V_0 = 2.12$, $\tau = 16$ ms, and $\tau_s = 4$ms.

The update of weights follows the below rules:

$$\Delta w_j = \begin{cases} \lambda_+ \sum_{t_i < t_{\max}} K(t_{\max} - t_i) & \text{if } P_+ \text{error}, \\ -\lambda_- \sum_{t_i < t_{\max}} K(t_{\max} - t_i) & \text{if } P_- \text{error}, \\ 0 & \text{otherwise}. \end{cases} \tag{8}$$

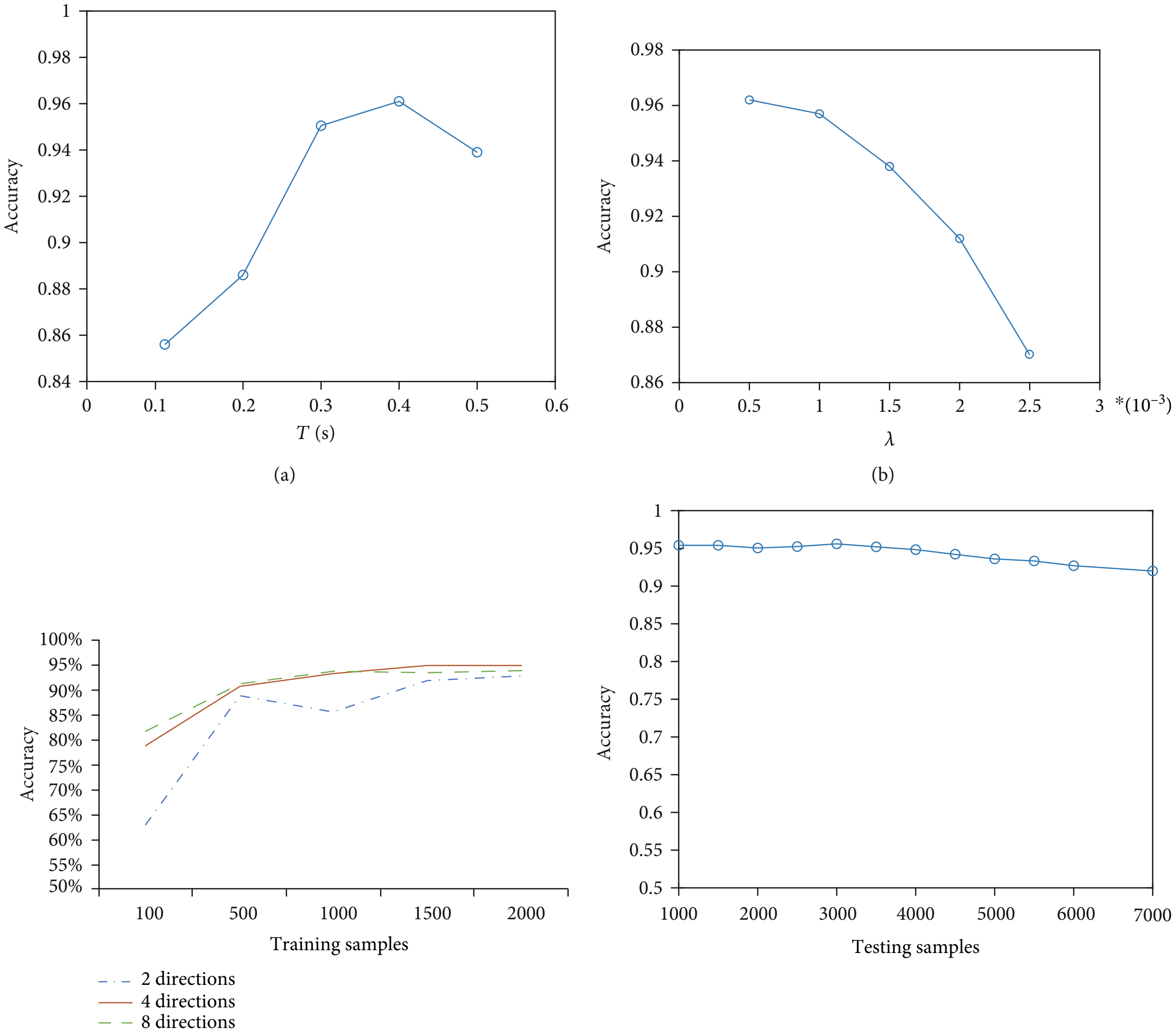
Here, $t_{\max}$ is the time when the membrane potential $V(t)$ of the decision layer reached the maximal value. The constant λ_+ and λ_- denotes the updating value of synaptic weights for each input spike. Δw_j is the change value of synaptic weight projecting to the jth output neuron, $j = 1, 2, ,\cdots, 10$. The classical tempotron rule is used in the binary classification problem. Each input pattern belongs to one of two classes (labeled P_+ and P_-). Assume that when input belongs to the P_+ (P_-) pattern, the output neuron should spike (not spike). But if the input belongs to P_+ and the output neuron does not spike, then P_+ is the error, so is the same for P_-. The tempotron learning is to modify the synaptic weights whenever an error occurs.

Here, we explore the tempotron learning for multiclassification task of the 10-digit recognition. During the training process, the correct output neuron which belongs to the current input pattern is labeled as P_+ and P_- for the other nine outputs. Therefore, the voltage V of the labeled neuron should pass the firing threshold for spiking, while the voltages V of the other nine neurons should be under the firing threshold (not spike). If this condition is not met, then the P_+ error and P_- error occurred. The weights will be updated until the outputs meet the condition. The average number of iterations for learning is usually around 5 for P_+ and 1 for P_-. The initial values of weights from the V2 layer to the decision layer are quite small (0.001), thus weights are gradually increased to trigger the correct output neuron firing.

7. Results

In order to evaluate the computational performance of our model, we applied it to the extensively studied MNIST database of handwritten digits. The simulation comprised three stages: a preprocessing stage in which the MNIST dataset was converted into orientation responses, a training phase, and a testing phase. The preprocessing stage was performed only once initially. Each image is spatially expanded into four pool orientation-selective responses of the V2 layer neurons, with each pool of neurons preferentially responded to one of four spatial orientations in 45° increments (horizontal "H," right diagonal "RD," vertical "V," and left diagonal "LD" (Figure 2(a)). After the single-spike temporal encoding, firing rates in the V2 layer are transformed into spike trains with each neuron that has only one spiking time as shown in Figure 2(b). Since the firing rate of each V2 neuron is encoded as a single spike with the spiking time proportional to the firing rate, the encoded spike trains of the four pools of orientation-selective neurons look like four triangles, where the image center of the four orientation population corresponding to the highest intensity of the V2 response shows the earliest encoded spiking time. Figure 2(c) shows the spiking neurons at three different times after the single-spiking temporal encoding, and we can see that center neurons with the highest firing rates fire at the earliest times, while edge neurons with low firing rates fire near the end of time window.

During the training stage, the change of synaptic weights Δw is calculated according to the kernel equation (8), where updating changes exponentially declined with the increase of time interval between the spiking time and the time when the membrane potential of the output layer reached the maximal value, i.e., $t_{\max}$. Since the number of presynaptic V2 spikings gradually increase during each time window, the output neurons reach the maximal value at the end of each time



(a)

(b)

(c)

(d)

Figure 3: The influence of the time window T (a) and weights updating amount λ (b) on the classification accuracy of our model. In this paper, we choose the optimal value as $T = 400$ms and $\lambda = 0.001$. (c) The influence of training samples on the computational performance of this model with 2, 4, or 8 directions in the V2 layer, where testing samples are 1000. (d) The influence of testing samples on the computational performance, where training samples are 2000.

Table 1: Comparisons with other spiking-based models on the performance of MNIST recognition.

Model	Train	Test	Accuracy (%)
Spiking deep CNN [16]	60000	10000	98.4
Spiking deep NN [30]	60000	10000	98.6
Spiking deep belief network [31]	60000	10000	94.09
SNN+NormAD [17]	50000	10000	98.2
Living NN+STDP [32]	10000	1000	74.7
SNN+STDP [24]	2000	1000	92
V2 layer+tempotron (this model)	2000	1000	96

window. Therefore, as shown in Figure 2(d), synaptic updates from center neurons to output neurons can be ignored due to the large time interval between their spiking times t_i and $t_{\max}$. Only weights from edge neurons to output neurons are significantly increased. By comparing Figure 2(c) and Figure 2(d), we can clearly see that firing activities of the edge neurons of each pool of the orientation population in V2 are quite similar with the weight matrix after learning. In this way, after training the temporal information of the V2 activity as shown in Figure 2(c), it is successfully transferred and saved as spatial information in the synaptic weights from V2 to the outputs as shown in Figure 2(d). For each sample image, synaptic weights are updated only once at the end of each time window. With the increase of training samples $n = 1,10,200$, more weights from edge

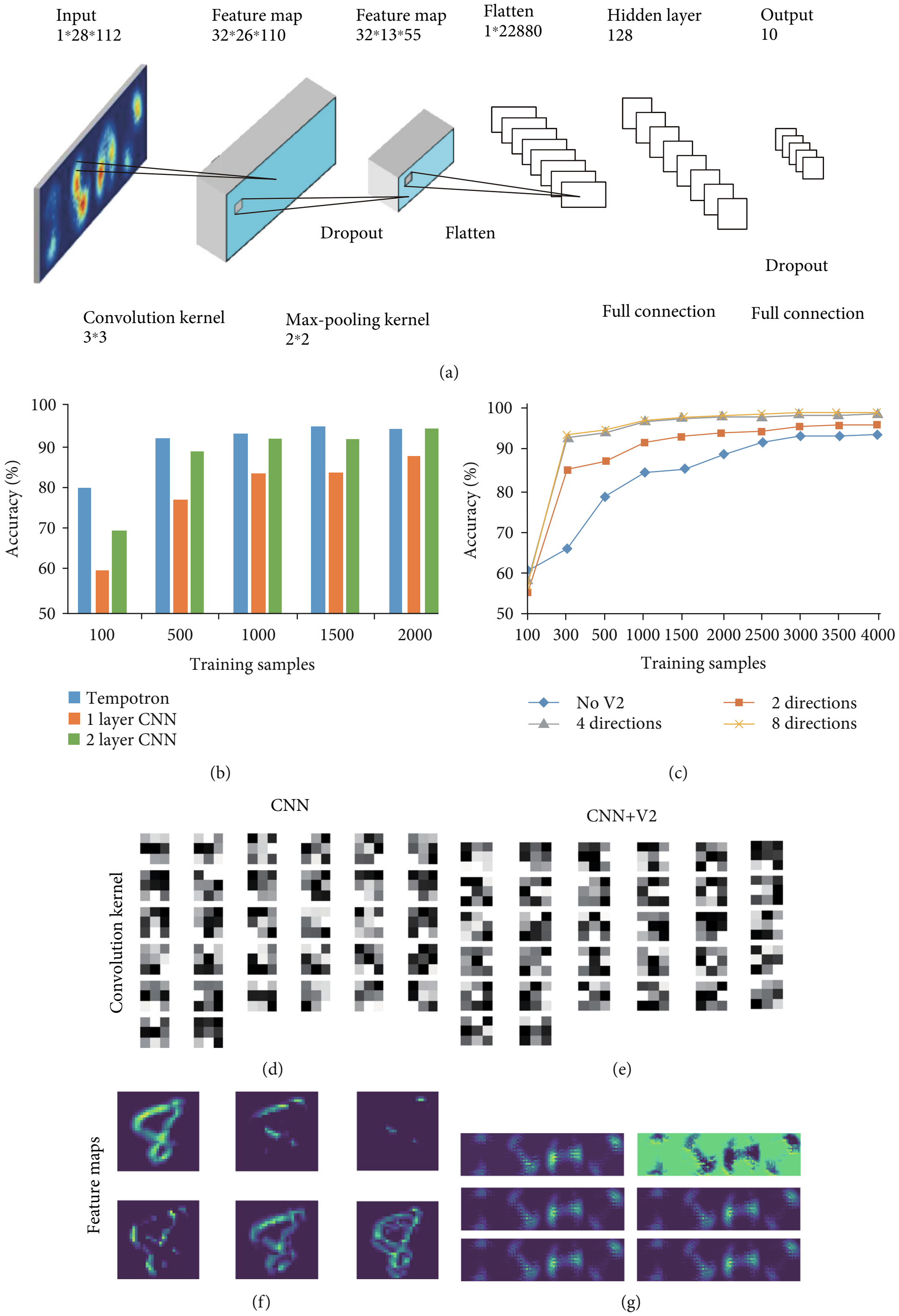


Figure 4: (a)The architecture of the convolutional neural network (CNN). The convolutional layer has 32 convolution kernels of size 3 × 3. The pooling layer has kernels of size 2 × 2. The output layer is a fully connected layer of 128 × 10, which is mapping to 10 classes. (b) The comparison of classification accuracy between our model and CNN with one or two convolutional layers in the tasks for different number of training samples. The testing sample is 1000. (c) The performance of 1 layer CNN by adding V2 layer with two, four, or eight directions before the convolutional layer. The two directions are horizontal and vertical while the four and eight directions are in 45° or 22 .5° increments, respectively. The convolutional kernels (d, e) and feature maps (f, g) for CNN (1 convolutional layer) with or without the V2 layer are compared.

neurons to the outputs are enhanced, making the weight image become more fuzzy and polarized.

Figures 2(e) and 2(f) illustrate a sample run for which the accumulation of voltage caused the labeled output neuron to cross the threshold. Since the encoded information of the presynaptic V2 firings are concentrated in the later part of time window as shown in Figure 2(b), the voltages rise from zero gradually during the whole time window. During the training process, the voltage of the labeled output neuron with input belongs to P_+ is accumulated repeatedly due to the increase of weights until it crosses the threshold. While the weights targeting to the other output neurons are almost not updated, keeping the other voltages under threshold. After training all the weights are fixed, the output neuron having the highest membrane voltage is taken as the final recognition result. From this figure, we can notice that for both training and testing process, the voltage of the correct output is significantly integrated and increased to reach to the threshold during the last 50ms of the time window. This means that only presynaptic spikes from the edge neurons which firing at the end of time window can be successfully propagated to the output layer. And only the correct output neuron connected by the previously learned weights from the corresponding four pools of V2 neurons receives the most information and shows the most rapid accumulation to reach to the threshold.

We also examine the influence of the parameter time window T and weights updating amount λ as shown in Figures 3(a) and 3(b). Due to the nonlinear dynamical characteristics of synaptic integration, if the time window T is too small, presynaptic information from V2 neurons cannot be completely propagated to the output layer, but if it is too large, since the synaptic current decays exponentially, it will lose much useful information after a long time of attenuation. Therefore, the value of T should be moderate to ensure that less information is lost in the process of synaptic transmission; otherwise, the performance of the network will be seriously reduced. In this paper, T is set to be the optimal value 400ms. For the parameter of weights updating amount λ, the updating time of synaptic weights will increase and prolong the training time when λ is too small, while if λ is too large, the synaptic weights will change greatly in each adjustment step, which will reduce the learning ability of the network and lead to the decline of recognition accuracy. Therefore, considering both of the recognition accuracy and time consumption, λ is set to be 0.001 in this paper.

Figure 3(c) demonstrates that the larger the number of orientations/directions in the V2 layer is, the smaller training samples are needed for achieving high recognition accuracy. For the cases of 4 directions and 8 directions, the accuracy can reach to 93% based on only 500 training samples. Besides, the computational performance of this model is very robust with the increase of testing samples as shown in Figure 3(d), with the training sample of 2000 for the 4 direction cases.

In Table 1, we compare our model with some other spiking-based neural networks on the task performance of the MNIST classification. Compared with the 60000 training samples and 10000 testing samples needed in the other methods, our model needs only 2000 training samples and 1000 testing samples and almost has no much loss of accuracy. In this paper, we used the tempotron learning rule to replace the STDP-like learning rule in [24], which further improved the classification accuracy on the MNIST dataset. The results show that the method adopted in this paper can achieve high classification performance with much smaller training samples.

Since convolutional neural networks (CNN) are most commonly used for image recognition in recent years, we specifically compared the computational accuracy of our model and traditional CNN with one or two convolutional layers in the task with different training sample numbers. The CNN architecture is shown in Figure 4(a). Figure 4(b) demonstrates that the accuracy increases with the increase of training samples for both our model and CNNs, but the performance of our model is always better than CNNs especially for small training samples. Noted that when training set is 500, the classification accuracy of our model achieves a high performance of 93% while the accuracy of CNN with two layers is only about 86%. Although CNN can increase the accuracy by adding convolution layers, it still cannot achieve good performance for small samples.

In order to examine whether the preprocessing of the V2 layer can improve the performance of CNN, we add the V2 layer with two, four, or eight directions before the convolutional layer of CNN (Figure 4(c)). The two directions are horizontal and vertical while the four and eight directions are in 45° or 22.5° increments, respectively. To avoid too much calculation, only CNN with one convolutional layer is examined by adding the V2 layer. From this figure, we can see that the V2 layer with four or eight directions can more significantly improve the performance of CNN for small training samples than the case of only two directions. When further increasing the number of training samples, the more training sets will lead to larger calculation cost but no significant improvement in the classification accuracy. Figures 4(d)–(g) compare the convolutional kernels and feature maps for CNN (one convolutional layer) with or without the conversion of V2 orientation responses from the image data. We can see that more feature maps for CNN with the V2 layer can be extracted than the traditional CNN, which contributes to the reduction of training samples and improvement of computational performance.

8. Conclusion

Here, we present a large-scale model of a hierarchical spiking neural network to perform a visual classification task in real-time. Based on the orientation-selective features of the primary visual cortical neurons and supervised tempotron learning method, we have successfully realized the high performance of MNIST handwriting digit recognition with small data samples. The orientation-selective layer expands the original image into high dimensional information to achieve highly detailed feature extraction, which provides the main contribution to reduce the training samples. Our results also show that the tempotron supervised learning rule is efficient to train the weight matrix between the V2 layer and output layer, which further improves the recognition accuracy. Compared with other image classification methods, our method needs much smaller data samples to achieve high performance of classification. Our results indicate a new brain-inspired computational model based on the visual cortex image processing mechanism to realize high-precision image recognition under small sample data. This work may provide a basis for the wide application of spiking neural networks in the field of intelligent computing, which is valuable and meaningful in both theoretical and applied researches.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflict of interest.

Acknowledgments

This work is supported by the Natural Science Foundation of Chongqing (No. cstc2019jcyj-msxmX0154).

References

[1] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun, “What is the best multi-stage architecture for object recognition?,” in *2009 IEEE 12th International Conference on Computer Vision*, pp. 2146–2153, Kyoto, Japan, 2009.

[2] P. A. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, pp. 511–518, Kauai, HI, USA, USA, 2001.

[3] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, “Frequency-tuned salient region detection,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1597–1604, Miami, FL, USA, 2019.

[4] B. Han, C. Yang, and X.-B. Gao, “Aurora Image Classification Based on LDA Combining with Saliency Information,” *Journal of Software*, vol. 24, no. 11, pp. 2758–2766, 2013.

[5] J. C. van Gemert, J.-M. Geusebroek, C. J. Veenman, and A. W. M. Smeulders, *Kernel Codebooks for Scene Categorization*, Lecture Notes in Computer Science, Springer, 2008.

[6] Y. Liao and V. R. Vemuri, “Use of K-Nearest Neighbor classifier for intrusion detection,” *Computers and Security*, vol. 21, no. 5, pp. 439–448, 2002.

[7] M. W. Gardner and S. R. Dorling, “Artificial neural networks (the multilayer perceptron): a review of applications in the atmospheric sciences,,” *Atmospheric Environment*, vol. 32, no. 14-15, pp. 2627–2636, 1998.

[8] J. A. K. Suykens and J. Vandewalle, “Least Squares Support Vector Machine Classifiers,” *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, 1999.

[9] Y. LeCun, B. Boser, J. S. Denker et al., “Backpropagation Applied to Handwritten Zip Code Recognition,” *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[10] D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, *Flexible, High Performance Convolutional Neural Networks for Image Classification*, International Joint Conference on Ijcai. DBLP, 2011.

[11] E. T. Rolls and G. Deco, *Computational neuroscience of vision*, Oxford University Press, Oxford, UK, 2002.

[12] Q. Yu and H. Tang, *Neuromorphic Cognitive Systems*, Springer International Publishing, Berlin, 2017.

[13] H. Markram, J. Lübke, M. Frotscher, and B. Sakmann, “Regulation of Synaptic Efficacy by Coincidence of Postsynaptic APs and EPSPs,” *Science*, vol. 275, no. 5297, pp. 213–215, 1997.

[14] S. Y. Kim and W. Lim, “Effect of Spike-Timing-Dependent Plasticity on Stochastic Burst Synchronization in A Scale-Free Neuronal Network,” *Cognitive Neurodynamics*, vol. 12, no. 3, pp. 315–342, 2018.

[15] X. Li, S. Luo, and F. Xue, “Effects of synaptic integration on the dynamics and computational performance of spiking neural network,” *Cognitive Neurodynamics*, vol. 14, no. 3, pp. 347–357, 2020.

[16] 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.

[17] S. R. Kulkarni and B. Rajendran, “Spiking neural networks for handwritten digit recognition-Supervised learning and network optimization,” *Neural Networks*, vol. 103, pp. 118–127, 2018.

[18] S. Kim and S. Park, “Spiking-YOLO: Spiking Neural Network for Energy-Efficient Object Detection,” 2019, https://arxiv.org/abs/1903.06530.

[19] T. Zhang, X. Pan, X. Xu, and R. Wang, “A cortical model with multi-layers to study visual attentional modulation of neurons at the synaptic level,” *Cognitive Neurodynamics*, vol. 13, no. 6, pp. 579–599, 2019.

[20] D. H. Hubel and T. N. Wiesel, “Receptive fields of single neurones in the cat's striate cortex,” *Journal of Physiology*, vol. 148, no. 3, pp. 574–591, 1959.

[21] E. P. Simoncelli and D. J. Heeger, “A model of neuronal responses in visual area MT,” *Vision Research*, vol. 38, no. 5, pp. 743–761, 1998.

[22] M. Beyeler, M. Richert, N. D. Dutt, and J. L. Krichmar, “Efficient Spiking Neural Network Model of Pattern Motion Selectivity in Visual Cortex,” *Neuroinformatics*, vol. 12, no. 3, pp. 435–454, 2014.

[23] M. Richert, J. M. Nageswaran, N. Dutt, and J. L. Krichmar, “An Efficient Simulation Environment for Modeling Large-Scale Cortical Processing,” *Frontiers in Neuroinformatics*, vol. 5, no. 19, 2011.

[24] 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 Networks*, vol. 48, no. 10, pp. 109–124, 2013.

[25] R. Gütig and H. Sompolinsky, “The tempotron: a neuron that learns spike timing-based decisions,” *Nature Neuroscience*, vol. 9, no. 3, pp. 420–428, 2006.

[26] E. M. Izhikevich, “Simple model of spiking neurons,” *IEEE Transactions on Neural Networks*, vol. 14, no. 6, pp. 1569–1572, 2003.

[27] E. M. Izhikevich, “Which model to use for cortical spiking neurons?,” *IEEE Transactions on Neural Networks*, vol. 15, no. 5, pp. 1063–1070, 2004.

[28] J. Moehlis, “Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting by Eugene M. Izhikevich,” *Siam Review*, vol. 50, no. 2, pp. 397–401, 2008.

[29] S. Panzeri, N. Brunel, N. K. Logothetis, and C. Kayser, “Sensory neural codes using multiplexed temporal scales,” *Trends in Neurosciences*, vol. 33, no. 3, pp. 111–120, 2010.

[30] P. U. Diehl, D. Neil, J. Binas, M. Cook, S.-C. Liu, and M. Pfeiffer, “Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing,” in *International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, Killarney, Ireland, July 2015.

[31] P. O'Connor, D. Neil, S.-C. Liu, T. Delbruck, and M. Pfeiffer, “Real-time classification and sensor fusion with a spiking deep belief network,” *Frontiers in Neuroscience*, vol. 7, no. 7, p. 178, 2013.

[32] Y. Zeng, K. Devincentis, Y. Xiao et al., “A Supervised Stdp-Based Training Algorithm for Living Neural Networks,” *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1154–1158, 2017.