Applied Artificial Intelligence

An International Journal

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/uaai20

R-STDP Based Spiking Neural Network for Human Action Recognition

S. Jeba Berlin & Mala John

To cite this article: S. Jeba Berlin & Mala John (2020) R-STDP Based Spiking Neural Network for Human Action Recognition, Applied Artificial Intelligence, 34:9, 656-673, DOI: 10.1080/08839514.2020.1765110

To link to this article: https://doi.org/10.1080/08839514.2020.1765110

Published online: 18 May 2020.

APPLIED ARTIFICIAL INTELLIGENCE
2020, VOL. 34, NO. 9, 656–673
https://doi.org/10.1080/08839514.2020.1765110

Taylor & Francis
Taylor & Francis Group

R-STDP Based Spiking Neural Network for Human Action Recognition

S. Jeba Berlin and Mala John

Department of Electronics Engineering, Madras Institute of Technology, Anna University, Chennai, India

ABSTRACT

Video surveillance systems are omnipresent and automatic monitoring of human activities is gaining importance in highly secured environments. The proposed work explores the use of the bio-inspired third generation neural network called spiking neural network (SNN) in order to recognize the action sequences present in a video. The SNN used in this work carries the neural information in terms of timing of spikes rather than the shape of the spikes. The learning technique used herein is reward-modulated spike time-dependent plasticity (R-STDP). It is based on reinforcement learning that modulates or demodulates the synaptic weights depending on the reward or the punishment signal that it receives from the decision layer. The absence of gradient descent techniques and external classifiers makes the system computationally efficient and simple. Finally, the performance of the network is evaluated on the two benchmark datasets, viz., Weizmann and KTH datasets.

ARTICLE HISTORY
Received 04 June 2019
Accepted 02 May 2020

Introduction

Human action recognition is considered as one of the active topics in computer vision and has many potential applications such as video surveillance (Vishwakarma and Agrawal 2013), social interaction modeling (Deng et al. 2016), human-computer interaction (Aggarwal and Ryoo 2011), sport event analysis, and health care (Gao et al. 2018). Different features that describe the human actions include appearance patterns (Cheng et al. 2015; Dalal, Triggs, and Schmid 2006), spatio-temporal interest points (Berlin and John 2016; Du et al. 2018), spatiotemporal body parts (Maity, Bhattacharjee, and Chakrabarti 2017), trajectories (Colque et al. 2017), skeletal joints (Kamel et al. 2019) and space–time gradients (Dalal and Triggs 2005). These are hand-crafted features tuned by human experts. Therefore, the features that give better performance in one application often result in poor performance in some other domain. Further, in some scenarios, there are challenges in discriminating different actions precisely. Recently, multi-view framework (Liu et al. 2015) has gained its popularity because it employs multiple cameras to tackle self-occlusion problems.

CONTACT S. Jeba Berlin jebaberlin@gmail.com Department of Electronics Engineering, Madras Institute of Technology, Campus of Anna University, Chennai-600044, India

Human actions possess high spatio-temporal complexity and long temporal correlation (Yao, Liu, and Huang 2016). So, it is preferable to extract the features in the spatiotemporal domain for the better discrimination of the intrinsic structure of the action sequences. But, it is rather difficult to configure a framework that exploits spatial & temporal information in parallel using the popular classifiers such as support vector machines, Bayes classifiers, Hidden Markov models (HMM) (Moayedi et al., 2016), and decision trees.

In contrast to traditional neural networks, spiking neural network (SNN) (Meng, Jin, and Yin 2011) is a powerful tool in exploiting the temporal information. In addition, hundreds of neurons are integrated into a single on-chip node, in order to improve parallel processing, communication cost, and energy saving (Xiang and Meng 2018). The SNN generates binary neural output pulses called spikes and the information regarding the action sequence is expressed in terms of the relative timing of the spikes rather than the shape of the spikes. In this work, the R-STDP-based SNN used for the task of object recognition task is re-purposed to the task of action recognition. Initially, the directional features are extracted using gradient filters of different orientations. Then, the synaptic weight is modulated using the learning algorithm called (R-STDP) (Mozafari et al. 2018). The R-STDP uses reinforcement learning where the polarity of the synaptic plasticity is updated based on the reward/punishment signal. Finally, the network learns the shape of the specific pattern of the action sequence when the same pattern is subjected to the network several times.

-The remainder of the work is organized as follows: The related works regarding human action recognition and spiking neural networks are discussed in section 2. The technical details of the proposed method are explained in section 3. The experimental results and discussions on the Weizmann dataset and KTH dataset are reported in section 4. The conclusion and the future scope of the work are explained in section 5.

Prior Works

Wang and Schmid (2013) proposed the model for human action recognition based on improved trajectories. The camera motion is eliminated by matching the points between frames using dense optical flow and SURF descriptors. Then, with these matches, the homography is estimated along with RANSAC. This method captures the appearance and motion information with motion-based descriptors such as histogram of optical flow and motion boundary histogram (MBH). Cheng et al. (2015) developed the human recognition method based on supervised temporal t-neighbor embedding for the human posture silhouette sequences of action frames. This method learns the explicit linear representations and the pattern manifold of the actions with the help of local neighbor relationship and linear projection.

Cao and Liu (2015) proposed type-2 fuzzy topic model, which assigns the topic labels for the mixture of action topics present in the video sequences. There are two membership functions; the first one is to measure the uncertainty of bag of words to the specific action topic and the next one is to evaluate the fuzziness of the first membership function. The slice-based representation described by Shan et al. (2015) delivers the temporal dynamics present in the spatio-temporal volume of the action sequence. The minimum average entropy principle is adopted to make the foreground pixels to be distributed only in the fewest slices, along the time axis. The MFCC feature extracted gives the spectral information regarding the temporal changes on the slice of variable lengths.

Al-Berry et al. (2016) represented the human actions based on 3D stationary wavelet coefficients and local binary pattern in order to take advantage of both local and global descriptors. It uses multiple classifiers and each classifier uses the features of different directional bands and the final decision is taken according to the votes for the particular sequence. This method is robust to scale variation and light changes but gives poor performance for the dynamic backgrounds. Liu et al. (2015) proposed the method that automatically learns the spatio-temporal action sequence using genetic programming (GP) with 3D Gabor and wavelet coefficients. The average cross-validation classification error calculated through the support vector machine is used as the GP fitness function. The color and optical flow features extracted by this method are robust to scale and shift variations.

Xu, Jiang, and Sun (2017) proposed two-stream dictionary-based learning for human action recognition. Here, the interest patches and their contour descriptors are computed for the spatial and temporal domain separately. Then, the dictionary is trained with these descriptors to create an action model. Finally, the score-based fusion is used to take the final decision on spatial and temporal classification results of the action sequence. Maity, Bhattacharjee, and Chakrabarti (2017) considered each frame as the human pose, and the histogram of the gradient is computed for every poses. It is then subjected to locality constrained linear coding to preserve the local details of the action and to get the sparse representation. The dictionary is formed for every video, and finally, the HMM is used for classification. This scheme preserves the spatial and temporal information providing regularized dimensionality reduction.

Different learning algorithms have been emerged to emulate the brain's computation for the processing of spatio-temporal patterns precisely. These learning rules are used to modify the synaptic weights of the neurons in spiking neural networks. The STDP-based unsupervised learning used by Yu et al. (2013) is the common learning algorithm, used and it considers the difference between the spiking time of pre-synaptic and post-synaptic neurons. The synaptic efficacy gets increased when the pre-synaptic spikes proceed the post-synaptic one and vice versa. Bohte, Kok, and La Poutre (2002) used the spikeprop-based supervised learning algorithm. It applies back propagation technique to update the synaptic weights and is as powerful as sigmoid neural networks.

Ponulak and Kasiński (2010) proposed ResuMe-based learning rule which is also based on supervised learning. It updates the weights based on the correlation between post and pre-spiking times. This method employs an error function which determines the time difference between the actual and desired spike trains. The PSD-based learning employed by Xu et al. (2018) modifies the synaptic weights in such a way that it reduces the error between the actual and desired output spikes. The membrane potential driven supervised learning method called MemPo Learn proposed by Zhang et al. (2018) adaptively changes the synaptic weights based on the difference between neuron membrane potential and its firing threshold. This works well even for the much smaller time steps. Here, the positive value of the error influences long-term potentiation, whereas the negative value causes long-term depression.

The SNN is considered to be the biologically plausible network and this has been now extended for the human action recognition systems. The dynamic evolving spiking neural network proposed by Dhoble et al. (2012) for human action recognition uses the combination of rank order spike coding and temporal spike coding and Fusis spike driven synaptic plasticity. It captures the spatiotemporal information effectively and fastly in an online mode. Jhuang et al. (2007) developed the HMAX mode which is the hierarchical feed-forward model that contains the hierarchy of convolutional (S) layers and max-pooling (C) layers. The convolution layer is a complex layer that computes the linear weighted sum of the inputs, whereas the pooling layer is a non-linear max layer that generates the shift- and scale-invariant features.

The gene regulatory network Bienenstock, Cooper, and Munro model is developed by Meng, Jin, and Yin (2011) for temporal pattern learning in human action recognition. The GRN used in this network regulates the synaptic plasticity and meta-plasticity that occur in the BCM-based SNN. The parameters in the GRN model are fine-tuned with the covariance matrix adaptation and with efficient evolutionary algorithms. Moreover, the corner-based spatial features are sensitive to noise and are scale variant. Liu et al. (2018) reported the multi-layer SNN (3D SNN) which has primary visual cortex and middle temporal cortex models to represent the motion features. The 3D Gabor filters and the 3D differences of Gaussian filters are employed for speed and direction selectivity, spatiotemporal inseparability, and center-surround suppression of neurons. The motion information is represented through spike trains, and finally, the SVM classifier is used for classifying the action sequences.

As evident from the recent literature, R-STDP based spiking network has not been applied to action recognition. Therefore, in this paper, the performance of the R-STDP spiking network-based framework for action recognition is discussed in detail.

Proposed Work

The spiking neural network consists of feature selective layer, local pooling layer, convolutional layer, and global pooling layer. The first layer is the feature extraction layer which employs a set of gradient filters of different orientations. For each frame, only the feature with the highest magnitude is considered for participation in the spike generation. It is then followed by local pooling layer to reduce the dimensionality and eliminate spatial redundancy within the neighborhood. The spike feature maps drawn from all the frames falling within the volume of action detection are fed simultaneously to the action recognition layer. The action recognition layer is based on reinforcement learning and this is the only layer subjected to synaptic weight updation. The overall structure of the proposed work is outlined in Figure 1.

Feature Selection Layer

In this layer, the ‘K’ input frames of ‘M’ video sequence are convolved with the gradient filters of four different orientations (0°, 45°, 90°, 135°) in order to extract the oriented edges. Thus, four feature maps are obtained from each frame of the action sequence. For every frame, at each pixel location, only the maximum magnitude of the feature value among the four feature maps is selected. This results in a single feature map, where the feature value at each pixel position is the winner of all the four feature maps. Therefore, ’K’ feature

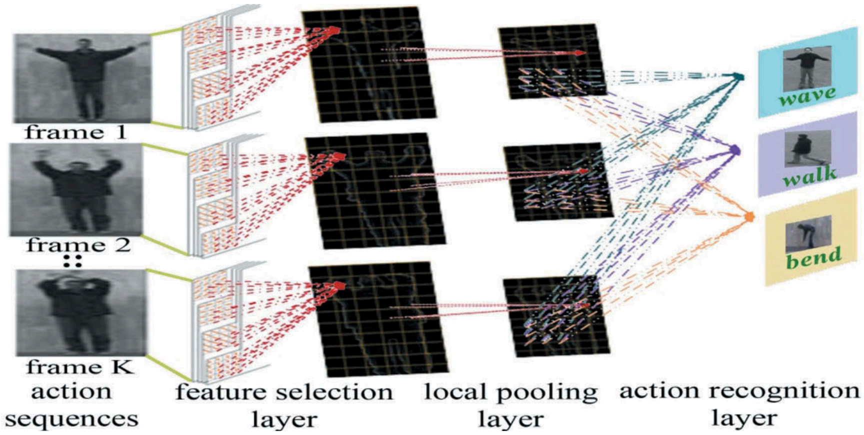


Figure 1. Structure of the proposed work.

maps are generated from 'K' input frames. For each frame of dimension, N × N will generate four feature vectors, giving rise to 4 × N × N points. Hence, each frame of dimension N × N will generate a spike feature map of dimension N × N.

The receptive fields used to obtain the oriented edges are presented in Figure 2. On convolving these receptive fields with the input frames, it generates four feature maps, each represents the edges on a particular orientation. Only, the absolute value of the feature maps is used for further analysis so as to give importance to the high contrast points.

The maximum value of the four different orientations at each pixel location is computed to produce a single feature map. It contains the winner of all the four features. Thus, it allows only one of the four orientations to fire at most once. Let $I_i^k(x, y)$ represents the feature component corresponding to the *ith* orientation of the *kth* frame; then, the competition between orientation is computed as

$$P^k(x, y) = \max_{i \in 1,2,3,4} (I_i^k(x, y)) \tag{1}$$

It is then followed by intensity to latency conversion which converts the feature values into the spike times. The spike time T(x, y) is inversely proportional to the feature value and is given by

$$T^k(x, y) = \begin{cases} \frac{1}{P^k(x,y)} & \text{for } P^k(x, y) \neq 0 \\ \eta & \text{otherwise} \end{cases} \tag{2}$$

Here $T^k(x, y)$ is the spike time feature map of size N × N, corresponding to the *kth* frame and η (fixed value) is the highest value, i.e., $\eta \gg T^k(x, y)$.

Local Pooling Layer

For the *kth* frame, $T^k(x, y)$, the spike time feature map is subjected to pooling. In this layer, the non-linear max pooling is performed over the set of neighboring neurons in order to gain the local invariance about the position of the edges. It is then followed by lateral inhibition. The spikes sustained from here are arranged in ascending order, i.e., the spike with the lowest latency is the first one to enter into the action detection layer.

-1	0	1
-2	0	2
-1	0	1

(a)

-2	-1	0
-1	0	1
0	1	2

(b)

-1	-2	-1
0	0	0
1	2	1

(c)

0	-1	-2
1	0	-1
2	1	0

(d)

Figure 2. Feature selective filters of different orientations (a) 0° (b) 45° (c) 90° (d) 135°.

The local pooling operation is performed over the window of size $q \times q$ with the stride of p = q-1. The spike with minimum latency within the specified window is allowed to propagate into the next layer. This eliminates the spatial redundancies in the visual information and thereby reduces the size of the subsequent layers. The output of the pooling layer, corresponding to the k^{th} frame, is denoted as Γ^k (x,y).

Action Recognition Layer

The output of all the 'K' frames, i.e., Γ^k(x,y), for all k = 1, 2, …, K, is fed to the action recognition layer in parallel. This is the only layer where the learning of action sequences, i.e., weight updation takes place. This layer receives input from the window of size r × r × K through the synapse connected to the local pooling layer. Thus, the membrane potential gets updated based on the magnitude of its synaptic weights on receiving each input spike. The neuron at this layer fires if the membrane potential reaches the thresholdλ. Finally, the synaptic weights get updated based on the reward/punishment signal generated from the subsequent layer. The algorithm used for weight updation is reward-modulated spike timing-dependent plasticity (R-STDP).

Spiking Neuron Training Phase

Assume there are 'C' classes. The synaptic weights of size r × r × K × Care initialized using Box–Mullerthe transform. The initial weight W_{ijkl} is generated following the normal distribution. Considering all the frames of the m^{th} action sequence, $\{\Gamma^k(x_j,y_j)\}$, k = 1, 2, K, the pooled spikes are arranged in ascending order. Only the lowest 'ρ' values of the spikes are considered for subsequent processing. The lowest values of spikes ($\Gamma^k(x_j,y_j)$), chosen for the m^{th} action sequence, are represented using the vector $\mathbf{S}^m$

$$\mathbf{S}^m = \left[s_1^m, s_2^m, s_3^m, \ \ldots \ , s_\rho^m\right] \tag{3}$$

where $s_t^m \neq 0$, $t \in 1,2, \ldots, \rho,, \rho$ is the number of spikes generated from the m^{th}action sequence and $s_1^m < s_2^m < s_3^m < \ \ldots \ < s_\rho^m$. Let the smallest spike correspond to the β^{th} frame and spatial location (x_{a1},y_{b1}), i.e., $s_1^n = \Gamma^\beta(x_{a1},y_{b1})$. The output membrane potential of the l^{th} neuron in the classification layer is computed as

$$U_{abkl} = \sum \sum\nolimits_{i,j\epsilon\ \psi_{ab}} W_{ijkl} \tag{4}$$

where ψ_{ab} represent the set of spikes over the window r × r centered around (x_a, y_b) of the kth frame. The weight $\{W_{ijkl}\}$, where i, j represent the spatial location of the connected pixels, k, the frame index and l, the action class of the output neuron is connected between convolutional and classification layers.

The membrane potentials are compared against the threshold λ. Out of the membrane potentials that exceeds the threshold, only the neuron corresponding to the highest potential is considered as the winner and chosen for firing. Let Ω represent the index of the winning neuron. As it is a supervised classification, all the synaptic weights connected to the Ω^{th}output neuron are updated as explained in the next section. This process is sequentially followed for each of the spikes, in the order of increasing latency.

Adaptive Learning Rules Based on R-STDP

The R-STDP proposed by Mozafari et al. (2018) is the learning rule used in this work for the updation of the synaptic weights. The neuron in the output layer that fires earlier is considered as the winner neuron and this is the one which involved in determining the network's decision. The synaptic weights receive the reward signal when the predicted output matches the actual output. Otherwise, it receives the punishment signal. The weight updation is carried out only for the synaptic weights that either receive the reward or punishment signal. In some cases, no signal is received and the weights are left undisturbed. The structure of the SNN and R-STDP-based weight updation is depicted in Figure 3. Consider the neuron in the classification layer is fired by the spike s_t^m, and Ω is the index of the winner neuron.

Case (i): The true class of the action sequence =Ω

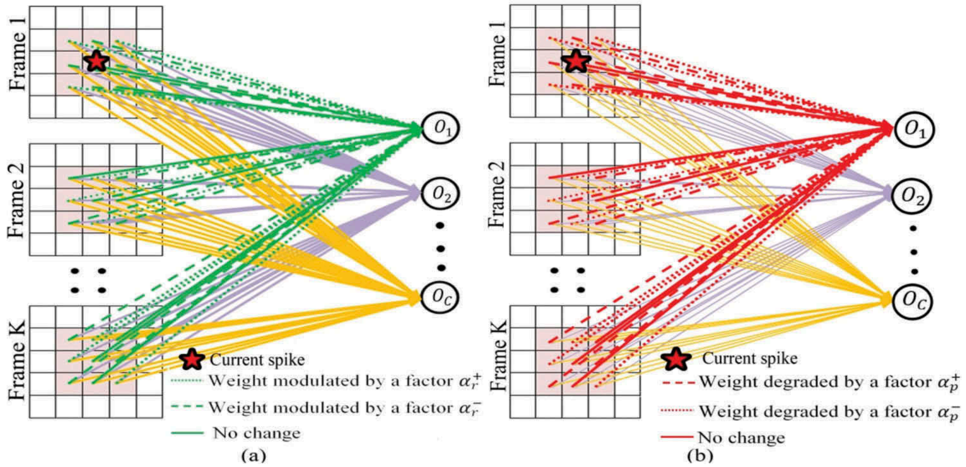

Frame 1
Frame 2
Frame K
O_1
O_2
O_C
Current spike
Weight modulated by a factor α_r^+
Weight modulated by a factor α_r^-
No change
(a)
Frame 1
Frame 2
Frame K
O_1
O_2
O_C
Current spike
Weight degraded by a factor α_p^+
Weight degraded by a factor α_p^-
No change
(b)

Figure 3. Weight updation in R-STDP SNN when the fired neuron in the classification layer is O_1 (a) on receiving reward signal (b) on receiving punishment signal.

The synaptic weights connected between Ω^{th} output neuron and r × r window centered around the spatial location corresponding to$\mathrm{s}_{\mathrm{t}}^{\mathrm{m}}$ are updated according to the following equations

$$\mathrm{W}_{\mathrm{ijk}\Omega(\mathrm{New})} = \begin{cases} \alpha_{\mathrm{r}}^{+} \times \mathrm{W}_{\mathrm{ijk}\Omega(\mathrm{old})} \times \left(1 - \mathrm{W}_{\mathrm{ijk}\Omega(\mathrm{old})}\right) & \mathrm{if}\Gamma^{\mathrm{k}}(\mathrm{i},\mathrm{j}) - \mathrm{s}_{\mathrm{t}}^{\mathrm{m}} \le 0 \\ \alpha_{\mathrm{r}}^{-} \times \mathrm{W}_{\mathrm{ijk}\Omega(\mathrm{old})} \times \left(1 - \mathrm{W}_{\mathrm{ijk}\Omega(\mathrm{old})}\right) & \mathrm{if}\Gamma^{\mathrm{k}}(\mathrm{i},\mathrm{j}) - \mathrm{s}_{\mathrm{t}}^{\mathrm{m}} > 0 \end{cases} \tag{5}$$

The constants$\alpha_{\mathrm{r}}^{+} > 0 and \alpha_{\mathrm{r}}^{-} > 0$, ($\alpha_{\mathrm{r}}^{+} > \alpha_{\mathrm{r}}^{-}$), are called as the learning coefficients.

Case (ii): The true class of the action sequence $\neq \Omega$

If the output neuron Ω is misclassified, the synaptic weights receive the punishment signal. The synaptic weights connected between Ω^{th} output neuron and each of the 'K' frames are degraded by a weight change given below

$$\mathrm{W}_{\mathrm{ijk}\Omega(\mathrm{New})} = \begin{cases} \alpha_{\mathrm{p}}^{+} \times \mathrm{W}_{\mathrm{ijk}\Omega(\mathrm{old})} \times \left(1 - \mathrm{W}_{\mathrm{ijk}\Omega(\mathrm{old})}\right) & \mathrm{if}\Gamma^{\mathrm{k}}(\mathrm{i},\mathrm{j}) - \mathrm{s}_{\mathrm{t}}^{\mathrm{m}} > 0 \\ \alpha_{\mathrm{p}}^{-} \times \mathrm{W}_{\mathrm{ijk}\Omega(\mathrm{old})} \times \left(1 - \mathrm{W}_{\mathrm{ijk}\Omega(\mathrm{old})}\right) & \mathrm{if}\Gamma^{\mathrm{k}}(\mathrm{i},\mathrm{j}) - \mathrm{s}_{\mathrm{t}}^{\mathrm{m}} \le 0 \end{cases} \tag{6}$$

The constants$\alpha_{\mathrm{p}}^{+} < 0 and \alpha_{\mathrm{p}}^{-} < 0$, ($|\alpha_{\mathrm{p}}^{+}| > |\alpha_{\mathrm{p}}^{-}|$), are called as the learning coefficients.

At the beginning of the training phase, the misclassification is relatively high since the weights are initialized randomly. When the number of training samples entering into the network increases, it subsequently increases the learning capability of the network. Also, the percentage of correctly classified samples gets increased. On receiving frequent punishment signals, there is a rapid decrease in the synaptic weights. But, the continuous incoming of reward signal strengthens the synaptic weights without bound. This results in overdetermination or underdetermination of synaptic weights.

The learning coefficients are learned adaptively to solve the overfitting problem. For this, the weight is modified for every epoch based on the number of correctly classified samples ($\mathrm{M}_{\mathrm{hit}}$) and the number of misclassified samples ($\mathrm{M}_{\mathrm{miss}}$). The learning coefficients are modified according to the equations given below

$$\begin{aligned} \alpha_{\mathrm{r(mod)}}^{+} &= \frac{\mathrm{M}_{\mathrm{miss}}}{\mathrm{M}} \alpha_{\mathrm{r}}^{+} \ \& \ \alpha_{\mathrm{r(mod)}}^{-} = \frac{\mathrm{M}_{\mathrm{hit}}}{\mathrm{M}} \alpha_{\mathrm{r}}^{-};\ \alpha_{\mathrm{p(mod)}}^{+} = \frac{\mathrm{M}_{\mathrm{miss}}}{\mathrm{M}} \alpha_{\mathrm{p}}^{+} \ \& \ \alpha_{\mathrm{p(mod)}}^{-} \\ &= \frac{\mathrm{M}_{\mathrm{hit}}}{\mathrm{M}} \alpha_{\mathrm{p}}^{-} \end{aligned} \tag{7}$$

where 'M' is the total number of action sequences used in the single epoch. At the end of the training process, every frame has a customized set of weight vectors which could be visualized as the convolutional mask of dimension r × r. This convolutional mask would be used during the testing process.

Results and discussion

The proposed work is implemented in python on Windows7 with the core i5 processor. The results are validated on two benchmark datasets and are explained in detail in the following sections. The results are evaluated for Weizmann (Blank et al. 2005) and KTH datasets (Laptev and Caputo 2004).

Initially, the frames are normalized to the size of 75 × 75 and passed through the Sobel gradient filters (given in Figure 2) of four different orientations ($0^\circ, 45^\circ$, 90°, 135°) to extract the local features present in the video frames. From the four feature maps generated, only the dominant intensity from each pixel location is retained and is allowed to transfer for the next level of processing. The features obtained through these filters are illustrated in Figure 4. Figure 4(a)–(e) represents the features along different orientations and Figure 4(f) represents the consolidated feature map which contains the winner feature of all the four feature maps. Thus, the four feature maps obtained are converted into a single feature map. Say if there are 20 frames in the action sequence, 20 feature maps each corresponds to different frames are generated. The feature values less than 0.0015 are set to be zero and the spikes are generated for the remaining values. The spike is the vector which stores information regarding the intensity value, spike time (inverse of intensity value), row coordinate, column coordinate, and the frame number at each pixel location.

The pooling layer uses the window of size 5 × 5 with the window stride equivalent to 4. After pooling, the size of the feature map is reduced to 25 × 25 × 20, where 20 represents the number of frames to be considered. Thus, it reduces the size of the feature map and also favors the rotation

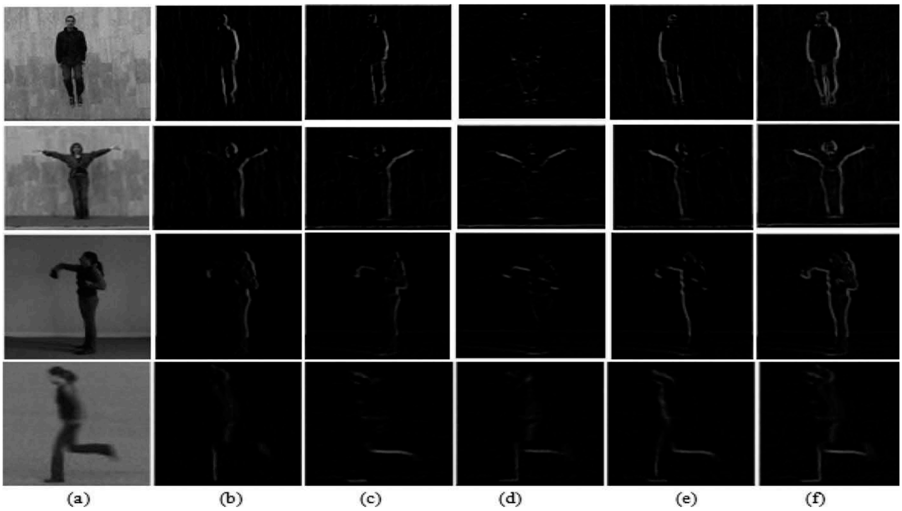


Figure 4. Images obtained through feature selective filters (a) input image (b) 0° (c) 45° (d) 90° (e) 135° (f) final feature map.

invariant property. The feature maps obtained through pooling followed by competition between spikes present in the frames (only alternate frames-1,3,5,7,9,11,13,15,17,19) are depicted in Figure 5(a)–(j). To have the visual representation of the motion sequences, the consolidated features of Figure 5 (a)–(j) are portrayed in Figure 5(k). The feature map of the proposed algorithm is visually similar to the motion history image (MHI) reported by Lin, Shao, and Lu (2016) to represent the human action based on the displacement of moving pixels. This demonstrates that the feature map delivers the local and global information regarding the actions. In addition, the MHI feature is robust to camera motion. Finally, the spikes are ordered in ascending order and are propagated sequentially to the next layer.

The feed-forward convolutional layer filters the features present in the feature maps of different frames based on the kernels and combine them in order to get a new representation. This uses integrate and fire model that integrates the features from different frames of the action sequences. The R-STDP employed here is used to update the synaptic weights during training. The synaptic weights are initialized randomly based on normal distribution of the mean ($\mu = 0.8$) and standard deviation ($\sigma = 0.005$). The pre-synaptic spikes are allowed to transfer into the next stage according to the order of its spiking time. At each time it receives the spike, the potential value at that particular position will get updated. At once the potential reaches the threshold value (λ), the current neuron is considered as the winning neuron. The most strongly activated neuron fires earlier followed by the less activated ones. Some less activated neurons will never take part in firing; thus, it reduces the computational complexity. Here, λ, the firing potential is fixed as $0.9 \times U_{max}$. Before initiating the weight updation

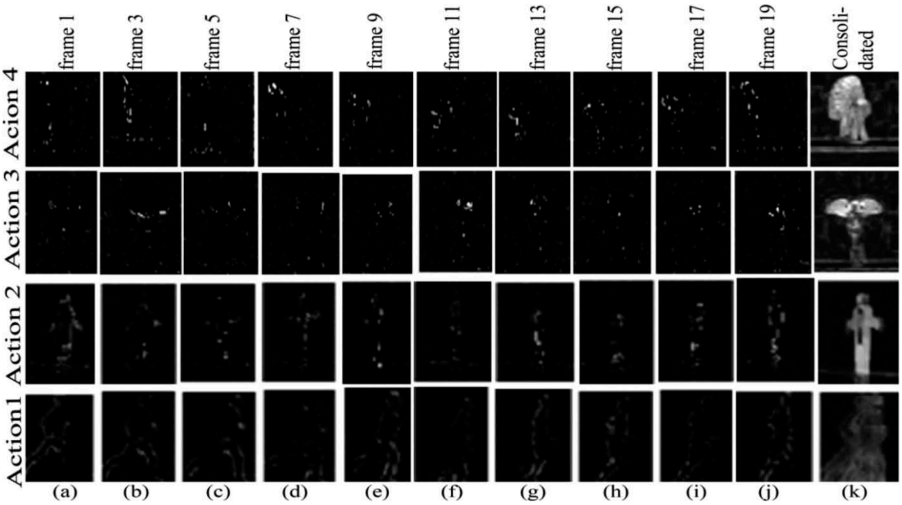


Figure 5. (a)–(j) Feature maps of different frames after pooling with lateral inhibition. (k) Consolidated features.

process, considering each of the ‘ρ’ spikes generated, corresponding to every action training sequence, the maximum output potential (U_{max}) is determined.

The learning coefficients are initialized as α_r^+=0.05, α_r^-=0.025,α_p^+ = 0.05 and α_p^-=0.005. The weights are then updated iteratively using Equations (5) and (6). At the end of every epoch, the learning coefficients are modified according to Equation (7). The updation of synaptic weights corresponding to different frames on various iterations is sketched in Figure 6. In order to get the visual representation of the weight changes, here, the size of the weight matrix is set to be the size of the pooled layer.

To avoid overfitting and underfitting problems, the learning coefficients are updated on every epoch according to the percentage of correctly classified samples and misclassified samples. The behavior of the spiking neural network on Weizmann dataset and KTH dataset is pictorialized in Figures 7 and 8. The network behaves in a chaotic fashion for early iterations and it can be easily spotted in the first few iterations. As the network continues its training, it becomes more selective to the particular pattern which makes the network stable. The network has the capacity to discriminate the features more quickly. This results in faster convergence of training samples. But, the testing sequences converge only in the latter iterations because of the adaptive learning rates. The network converges with a constant rate even after the sequences are classified correctly because the learning coefficients are not allowed to drop below 20% from its initial value.

The performance of the proposed work is analyzed on Weizmann and KTH dataset. The Weizmann dataset contains 10 actions: bend (BD), jack (JK), jump (JP), pjump (PP), run (RN), side (SD), skip (SP), walk (WK),

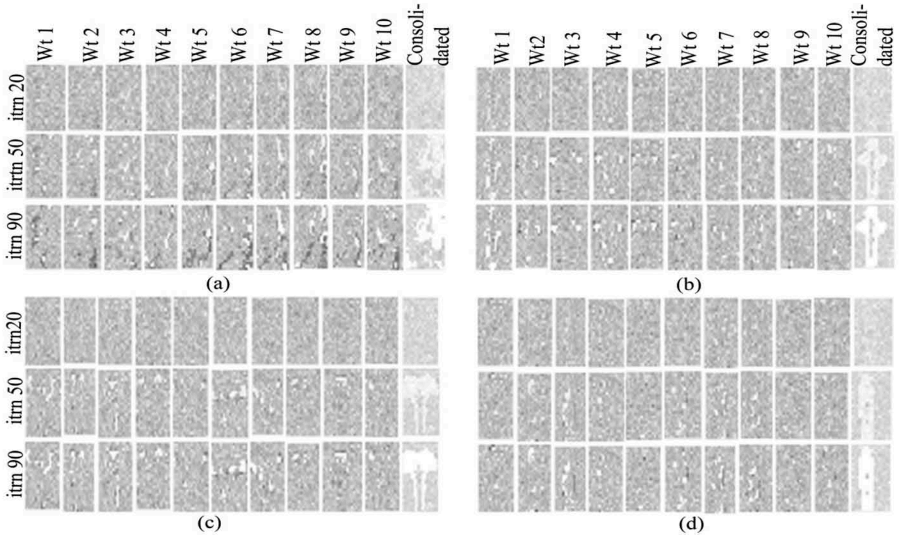


Figure 6. Evolution of different features from the video frames on various iterations.

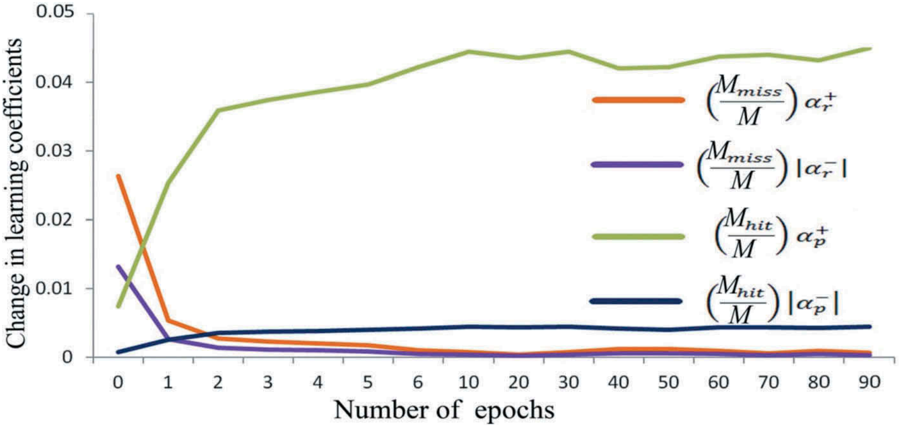


Figure 7. Trajectory of changes in learning rate on Weizmann dataset with respect to the probability of correct and incorrect classification.

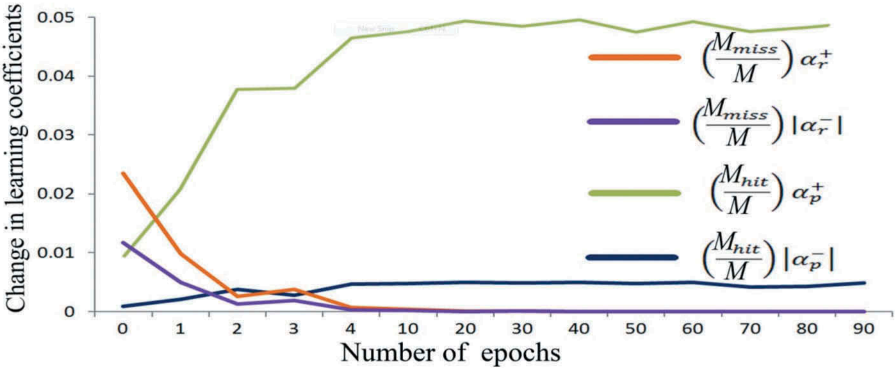


Figure 8. Trajectory of changes in learning rate on KTH dataset with respect to the probability of correct and incorrect classification.

wave1 (W1), and wave2 (W2) with static background. The Leave one out cross-validation technique is used for the evaluation of the Weizmann database. The KTH dataset consists of six types of human actions: walking (WK), jogging (JG), running (RN), boxing (BX), hand waving (HW), and hand clapping (HC) taken over homogenous background in four different scenarios. In this data set, 70% of the samples are considered as the training set and the remaining 30% of the samples are used for testing. The blank frames in the KTH datasets are removed in order to get rid of useless information that produces a negative impact on recognition results.

The training and testing accuracies on Weizmann and KTH dataset are depicted in Figure 9. The training and testing accuracies are gradually increased till 40^{th} to 50^{th} epochs. The training samples get converged to the classification accuracy of 100% when it crosses 60^{th} epoch. In case, when the network is subjected to more number of repeated training examples, the network converges

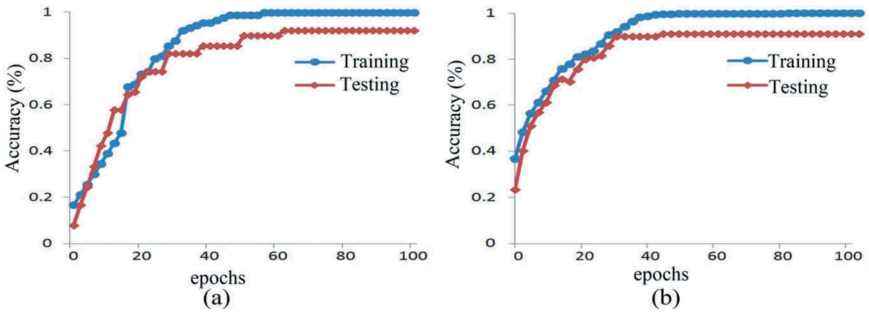
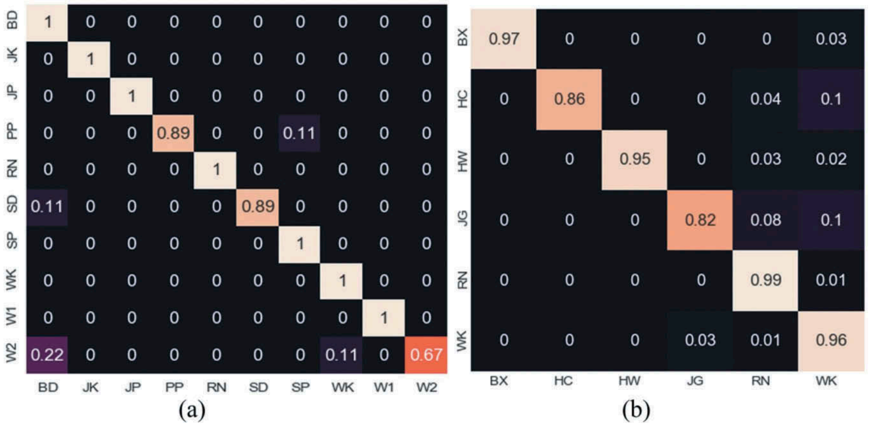


Figure 9. Training and Testing accuracies on (a) Weizmann dataset (b) KTH dataset.

faster than the earlier. But, it suffers from overfitting problem, and in turn, it lacks the generalization capability.

The confusion matrix obtained on Weizmann and KTH datasets is shown in Figure 10. From the figure, it is shown that the actions such as bend (BD), jack (JK), jump (JP), run (RN), skip (SP), walk (WK), and one-handed waving (W1) on Weizmann dataset are high upto 100%. The action 'handwave' works excellently with 99.9% classification rate on KTH dataset. Due to the presence of similar motion patterns for different actions, those actions are often misclassified as one of the other action sequences.

The proposed work is compared with the existing methods that use the same database and apply spiking neural networks based approaches. The accuracies obtained for the biologically inspired methods for human action recognition are demonstrated in Table 1. The proposed work achieves the performance of 94.44% and 92.50% on Weizmann and KTH dataset which is higher than Meng's. The GRN-BCM-based SNN model proposed by Meng et al., 2010 considered only the spatial features. The performance of the proposed work is comparable to the methods proposed by Jhuang et al., 2007 and Escobar et al., 2012.



BD	1	0	0	0	0	0	0	0	0	0
JK	0	1	0	0	0	0	0	0	0	0
JP	0	0	1	0	0	0	0	0	0	0
PP	0	0	0	0.89	0	0	0.11	0	0	0
RN	0	0	0	0	1	0	0	0	0	0
SD	0.11	0	0	0	0	0.89	0	0	0	0
SP	0	0	0	0	0	0	1	0	0	0
WK	0	0	0	0	0	0	0	1	0	0
W1	0	0	0	0	0	0	0	0	1	0
W2	0.22	0	0	0	0	0	0	0.11	0	0.67
	BD	JK	JP	PP	RN	SD	SP	WK	W1	W2

(a)

BX	0.97	0	0	0	0	0.03
HC	0	0.86	0	0	0.04	0.1
HW	0	0	0.95	0	0.03	0.02
JG	0	0	0	0.82	0.08	0.1
RN	0	0	0	0	0.99	0.01
WK	0	0	0	0.03	0.01	0.96
	BX	HC	HW	JG	RN	WK

(b)

Figure 10. Confusion matrix on (a) Weizmann dataset (b) KTH dataset.

Table 1. Performance comparison of existing methods.

Authors	Weizmann	KTH
Jhuang et al. (2007)	96.30	91.70
Escobar et al. (2012)	99.26	92.44
Meng et al. (2010)	-	82.50
Meng, Jin, and Yin (2011)	74.44	84.81
Liu et al. 2018	98.52	93.16
Proposed work	94.44	92.50

The Gaussian mixture models (GMM) based background subtraction and optical flow-based motion features used by Jhuang et al. are computationally intensive. The performance of the proposed work (92.50%) is superior (91.70%) on the KTH action dataset. The method reported by Escober et al. has the difficulties to choose the best spatio-temporal bandwidth for the energy filters and to obtain the self-centered representation of the action sequence. They have also excluded the running sequence of the action from the dataset to get a better response. The computational cost (0.02 s) of the proposed method is low compared to the method proposed by Liu et al. (30 s). Besides, all the above methods need external classifier for classification.

However, the proposed work does not require any pre-processing steps to extract the spatial and temporal information from the action sequences. This work does not possess any additional classifiers for classification. The model converges as earlier as 100th epoch. Also, this method is computationally simple with only one trainable layer, and hence, it takes less time for training. The time taken for training and testing the model on Weizmann and KTH dataset is 245.84 s and 3651.84 s, respectively. The time taken for testing the sample is 0.02 s.

Conclusion

The proposed work is done on human action recognition using R-STDP based spiking neural network. The information is encoded into the spike times using the temporal coding scheme. The R-STDP based spiking neural network used in this work performs both the feature extraction and classification in an end to end manner. The framework is very simple as it involves 4D feature extraction layers with fixed convolutional masks and a single network layer which requires training. Since only the earliest spike took part in taking network's decision, this method is computationally simple, is energy efficient, and is the suitable candidate for the hardware implementations. In the future, the accuracy of the proposed work could be further improved by adding deep learning models, at an increased computational cost. Furthermore, this work would be extended to suit for the real-time scenarios.

ORCID

S. Jeba Berlin http://orcid.org/0000-0002-7633-6083
Mala John http://orcid.org/0000-0001-5034-3405

References

Aggarwal, J. K., and M. S. Ryoo. 2011. Human activity analysis: A review. *ACM Computing Surveys (CSUR)* 43 (3):16. doi:10.1145/1922649.1922653.

Al-Berry, M. N., M. A. M. Salem, H. M. Ebeid, A. S. Hussein, and M. F. Tolba. 2016. Fusing directional wavelet local binary pattern and moments for human action recognition. *IET Computer Vision* 10 (2):153–162. doi: 10.1049/iet-cvi.2015.0087.

Berlin, S. J., and M. John. 2016, October. Human interaction recognition through deep learning network. In 2016 IEEE International Carnahan Conference on Security Technology (ICCST) (pp. 1–4). Orlando, FL: IEEE.

Blank, M., L. Gorelick, E. Shechtman, M. Irani, and R. Basri. 2005, October. Actions as space-time shapes. In Tenth IEEE International Conference on Computer Vision (ICCV'05) (Vols. 1, 2, pp. 1395–402). Beijing, China: IEEE.

Bohte, S. M., J. N. Kok, and H. La Poutre. 2002. Error-backpropagation in temporally encoded networks of spiking neurons. *Neurocomputing* 48 (1–4):17–37. doi:10.1016/S0925-2312(01)00658-0.

Cao, X.-Q., and Z.-Q. Liu. 2015. Type-2 fuzzy topic models for human action recognition. *IEEE Transactions on Fuzzy Systems* 23 (5):1581–93. doi:10.1109/TFUZZ.2014.2370678.

Cheng, J., H. Liu, F. Wang, H. Li, and C. Zhu. 2015. Silhouette analysis for human action recognition based on supervised temporal t-SNE and incremental learning. *IEEE Transactions on Image Processing* 24 (10):3203–17. doi:10.1109/TIP.2015.2441634.

Colque, R. V. H. M., C. Caetano, M. T. L. de Andrade, and W. R. Schwartz. 2017. Histograms of optical flow orientation and magnitude and entropy to detect anomalous events in videos. *IEEE Transactions on Circuits and Systems for Video Technology* 27 (3):673–82. doi:10.1109/TCSVT.2016.2637778.

Dalal, N., and B. Triggs. 2005, June. Histograms of oriented gradients for human detection. In International Conference on Computer Vision & Pattern Recognition (CVPR'05) (Vol. 1, pp. 886–93). San Diego, CA: IEEE Computer Society.

Dalal, N., B. Triggs, and C. Schmid. 2006, May. Human detection using oriented histograms of flow and appearance. In European Conference on Computer Vision (pp. 428–41). Springer, Berlin, Heidelberg.

Deng, Z., A. Vahdat, H. Hu, and G. Mori. 2016. Structure inference machines: Recurrent neural networks for analyzing relations in group activity recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4772–81). Las Vegas, NV.

Dhoble, K., N. Nuntalid, G. Indiveri, and N. Kasabov. 2012, June. Online spatio-temporal pattern recognition with evolving spiking neural networks utilising address event representation, rank order, and temporal spike learning. In The 2012 International Joint Conference on Neural networks (IJCNN) (pp. 1–7). Brisbane, QLD, Australia: IEEE.

Du, W., Y. Wang, and Y. Qiao. 2018. Recurrent spatial-temporal attention network for action recognition in videos. *IEEE Transactions on Image Processing* 27 (3):1347–1360.

Escobar, M.-J., and P. Kornprobst. 2012. Action recognition via bio-inspired features: The richness of center–surround interaction. *Computer Vision and Image Understanding* 116 (5):593–605. doi:10.1016/j.cviu.2012.01.002.

Gao, Y., X. Xiang, N. Xiong, B. Huang, H. J. Lee, R. Alrifai, X. Jiang, and Z. Fang. 2018. Human action monitoring for healthcare based on deep learning. *IEEE Access* 6:52277–85. doi:10.1109/ACCESS.2018.2869790.

Jhuang, H., T. Serre, L. Wolf, and T. Poggio. 2007. A biologically inspired system for action recognition. In IEEE International Conference on Computer Vision, (p. 1). Rio de Janeiro, Brazil.

Kamel, A., B. Liu, P. Li, and B. Sheng. 2019. An investigation of 3D human pose estimation for learning Tai Chi: A human factor perspective. *International Journal of Human–Computer Interaction* 35 (4–5):427–39. doi:10.1080/10447318.2018.1543081.

Laptev, I., and B. Caputo. 2004, August. Recognizing human actions: A local SVM approach. In International conference on Pattern Recognition (pp. 32–36). Cambridge, UK: IEEE.

Liu, A.-A., Y.-T. Su, P.-P. Jia, Z. Gao, T. Hao, and Z.-X. Yang. 2015. Multiple/single-view human action recognition via part-induced multitask structural learning. *IEEE Transactions on Cybernetics* 45 (6):1194–208. doi:10.1109/TCYB.2014.2347057.

Liu, H., N. Shu, Q. Tang, and W. Zhang. 2018. Computational model based on neural network of visual cortex for human action recognition. *IEEE Transactions on Neural Networks and Learning Systems* 29 (5):1427–40. doi:10.1109/TNNLS.2017.2669522.

Liu, L., L. Shao, X. Li, and K. Lu. 2016. Learning spatio-temporal representations for action recognition: A genetic programming approach. *IEEE Transactions on Cybernetics* 46 (1):158–70. doi:10.1109/TCYB.2015.2399172.

Maity, S., D. Bhattacharjee, and A. Chakrabarti. 2017. A novel approach for human action recognition from silhouette images. *IETE Journal of Research* 63 (2):160–71. doi:10.1080/03772063.2016.1242383.

Meng, Y., Y. Jin, and J. Yin. 2011. Modeling activity-dependent plasticity in BCM spiking neural networks with application to human behavior recognition. *IEEE Transactions on Neural Networks* 22 (12):1952–66. doi:10.1109/TNN.2011.2171044.

Meng, Y., Y. Jin, J. Yin, and M. Conforth. 2010, July. Human activity detection using spiking neural networks regulated by a gene regulatory network. In The 2010 International Joint Conference on Neural Networks (IJCNN) (pp. 1–6). Barcelona, Spain: IEEE.

Moayedi, F., Z. Azimifar, and R. Boostani. 2016. Human action recognition: learning sparse basis units from trajectory subspace. *Applied Artificial Intelligence* 30 (4):297–317. doi:10.1080/08839514.2016.1169094.

Mozafari, M., S. R. Kheradpisheh, T. Masquelier, A. Nowzari-Dalini, and M. Ganjtabesh. 2018. First-spike-based visual categorization using reward-modulated STDP. *IEEE Transactions on Neural Networks and Learning Systems* (99):1–13. doi:10.1109/TNNLS.2018.2826721.

Ponulak, F., and A. Kasiński. 2010. Supervised learning in spiking neural networks with ReSuMe: Sequence learning, classification, and spike shifting. *Neural Computation* 22 (2):467–510. doi:10.1162/neco.2009.11-08-901.

Shan, Y., Z. Zhang, P. Yang, and K. Huang. 2015. Adaptive slice representation for human action classification. *IEEE Transactions on Circuits and Systems for Video Technology* 25 (10):1624–36. doi:10.1109/TCSVT.2014.2376136.

Vishwakarma, S., and A. Agrawal. 2013. A survey on activity recognition and behavior understanding in video surveillance. *The Visual Computer* 29 (10):983–1009. doi:10.1007/s00371-012-0752-6.

Wang, H., and C. Schmid. 2013. Action recognition with improved trajectories. In Proceedings of the IEEE International Conference on Computer Vision. (pp. 3551–58). Sydney, NSW, Australia.

Xiang, Y., and J. Meng. 2018. A cross-layer based mapping for spiking neural network onto network on chip. *International Journal of Parallel, Emergent and Distributed Systems* 33 (5):526–44. doi:10.1080/17445760.2017.1399206.

Xu, K., X. Jiang, and T. Sun. 2017. Two-stream dictionary learning architecture for action recognition. *IEEE Transactions on Circuits and Systems for Video Technology* 27 (3):567–76. doi:10.1109/TCSVT.2017.2665359.

Xu, X., X. Jin, R. Yan, Q. Fang, and W. Lu. 2018. Visual pattern recognition using enhanced visual features and PSD-based learning rule. *IEEE Transactions on Cognitive and Developmental Systems* 10 (2):205–12. doi:10.1109/TCDS.2017.2769166.

Yao, L., Y. Liu, and S. Huang. 2016. Spatio-temporal information for human action recognition. *EURASIP Journal on Image and Video Processing* 2016 (1):39. doi:10.1186/s13640-016-0145-2.

Yu, Q., H. Tang, K. C. Tan, and H. Li. 2013. Rapid feedforward computation by temporal encoding and learning with spiking neurons. *IEEE Transactions on Neural Networks and Learning Systems* 24 (10):1539–52. doi:10.1109/TNNLS.2013.2245677.

Zhang, M., H. Qu, A. Belatreche, Y. Chen, and Z. Yi. 2018. A highly effective and robust membrane potential-driven supervised learning method for spiking neurons. *IEEE Transactions on Neural Networks and Learning Systems* (99):1–15. doi:10.1109/TNNLS.2018.2833077.