Spiking Emotions: Dynamic Vision Emotion Recognition Using Spiking Neural Networks

Binqiang Wang[1,2], Gang Dong[2*], Yaqian Zhao[2], Rengang Li[2], Hongbin Yang[2], Wenfeng Yin[2], Lingyan Liang[2]

[1]*Shandong Massive Information Technology Research Institute, China*
[2]*State Key Laboratory of High-end Server & Storage Technology Inspur (Beijing) Electronic Information Industry Co., Ltd. Beijing, China*

Abstract

Emotion recognition from vision information is a significant research topic in the computer vision community. The current prevalent solution based on Artificial Neural Networks (ANNs) demonstrates high accuracy but large computation consumption. Compared with ANNs, Spiking Neural Networks (SNNs) are more biologically realistic and computationally effective. However, it still remains a great challenge to utilize SNNs to vision emotion recognition, mainly due to the lack emotional dataset of Dynamic Vision Sensor (DVS) and a properly designed SNN framework. In this paper, we present a method to generate a simulation dataset of DVS, leveraging the existed emotion recognition dataset containing video segments. Meanwhile, an SNN framework and its counterpart ANNs are adopted to complete dynamic vision emotion recognition based on the simulated DVS dataset and original frames data respectively. The proposed SNN framework consists of a feature extraction module to extract informative features based on spike-trains of input, a voting neurons group module containing two groups of emotional neurons, and an emotional mapping module to translate output spike-trains to emotion polarity labels. The results demonstrate that compared with the ANN, the proposed SNN can achieve better performance and its energy consumption is only one-quarter of the ANN's.

Keywords

spiking neural network; dynamic vision sensor; emotion recognition

1. Introduction

Emotion recognition, as a hot research topic in the affective computing community, has derived many researchers' attention coming from domains like computer vision [1, 2], natural language processing [3], speech processing [4, 5], and human-computer interaction [6, 7]. At present, most methods adopt Artificial Neural Networks (ANNs) to perform emotion recognition, which achieves state-of-art solutions. An efficient emotion recognition method will facilitate communication between people on the wearable scenes [8]. However, the high energy consumption of ANNs hinders emotion recognition's application on embedded and mobile devices. Although knowledge distillation [9] and neural architecture search [10] can obtain ANN architecture with fewer parameters to reduce energy consumption and be suitable for mobile devices, it does not change the essence of ANNs.

As the third generation of neural networks, Spiking Neural Network (SNN) [11] with low power consumption is one potential solution to lead to an embedded and mobile emotion recognition algorithm reality. Some researches applying SNNs to complete emotion recognition tasks have been proposed to extract emotion information from speech, cross-modal, or electroencephalograph (EGG) [12, 13, 14, 15, 16]. The feature extraction in most of these methods involves the pre-processing operation, the audio feature extraction such as Mel cepstrum coefficient. To complete emotion recognition, a shallow SNN, a three-layer in most existing methods is adopted as a classifier. Based on these techniques, previous methods have accomplished encouraging performance on relevant datasets. Nevertheless, it remains

AHPCAI2022@2nd International Conference on Algorithms, High Performance Computing and Artificial Intelligence
EMAIL: * Corresponding author: donggang@inspur.com (Gang Dong)

challenging to extract emotional representative information using SNNs from video segments. The first challenge is to collect an emotion recognition dataset utilizing a dynamic vision sensor, which is expensive to conduct. To mitigate this cost, a simulated method to generate simulated spikes-like data herein is proposed inspired by frame difference encoding in [17]. Note that in order to better simulate the mechanism of human ocular nerve receiving information, a kind of float value frame is adopted in the simulated method, which is a novel scheme in the spiking encoding domain [18, 19]. On the other hand, the structures in existing SNN-based emotion recognition methods are simple and the spiking neuron model used in most previous literature is Leaky Integrate-and-Fire (LIF) model. To take full advantage of existing abstract structures in ANNs, a framework is designed to leverage the latest progress of SNN proposed in [20], where a new spiking neuron model is termed Parametric Leaky Integrate-and-Fire (PLIF) neuron model.

In this paper, we propose a scheme that is capable of combining the advantage of short-term high-performance results based on ordinary cameras and the low energy consumption of dynamic vision sensors. Thus, the simulated data contains both float-value data in the first capture of the scene and spike trains data during the remaining observation period. Experiments are designed to demonstrate the effectiveness of the proposed scheme.

Our contributions are summarized as:

1) To the best of our knowledge, this is the first attempt to apply the SNNs in emotion recognition based on simulated dynamic vision sensor data. As SNNs have higher biological plausibility compared with ANNs, the combination of SNNs and the dynamic vision sensor may be supportive to exploit the emotion possessed by humans.
2) We propose a method to generate simulated dynamic vision sensor data. Note that the generated data is not pure spikes. Considering the real application scene, the first frame data is represented by float-value and the following frames consist of pure spikes.
3) Parametric Leaky Integrate-and-Fire (PLIF) is adopted to construct the SNN in this paper. We evaluate the SNN on the simulated dynamic vision sensor data. The SNN achieves better performance compared with the counterpart ANN.

2. Method

2.1 DVS Simulation Algorithm

The simulation algorithm is explained in detail in this section. Firstly, the concept of DVS is introduced briefly. Then, the data format of the DVS is clarified. Finally, the simulation algorithm is present to generate DVS format data based on video segments.

Focusing on the dynamic information, DVS records the dynamic changes of a scene that is under perception. Different from recording the whole scene pixel by pixel with a float number representing the intensity of light in traditional cameras, DVS only captures the changes of light of the scene, the recorded contents are either 0 or 1, which indicates whether the intensity of a location in the scene has changed. It is not trivial to directly collect emotion recognition data using DVS as DVS is expensive. An alternative idea is to generate simulated dynamic vision emotion recognition data in terms of the data format and the existing vision emotion dataset.

The data generated by DVS are named neuromorphic data which is represented by $\mathrm{E}(x_i, y_i, t_i, p_i)$ (i=0,1,...,N-1), where x_i, y_i is the location where the event happened, t_i is the time when the event occurred, p_i is the polarity of the event.

Emotion recognition based on video segments provides a series of frames that record the change in a scene, which is publicly available [21]. To simulate a DVS's output based on them, the RGB frame is converted to gray to represent the intensity of each frame. Then the difference between adjoin frames is used to generate the polarity. A hyperparameter is named sensitivity to represent the degree of the intensity change. Finally, the spiking is formed by frame series order in the original video to the simulation output. The final representation is the simulation of DVS's output based on video segments, which is compatible with the data format recorded by DVS. Note that to consider the real phenomenon: We catch a scene firstly with whole and then attend to the change. Inspired by this phenomenon, the first frame is set as float-valued gray information. In other words, the output of the algorithm includes two parts: the first frame with float-value representing the ordinary camera and other frames of spike values representing dynamic vision sensor.

2.2 Neuron Model

The principle of SNN is to mimic the cell in the brain on the micro-physiological scale. So the neuron model in SNN is different from that in traditional ANNs. Generally, the basic neuron model in ANN is the McCulloch-Pitts model, while the popular component neuron model in SNN is Leaky Integrate-and-Fire (LIF) model. The information transition in neuron cells is not just the summation of all the input coming from other neurons by synapses. Actually, as time goes on, the input is accumulated in the cell membrane to cause the increase of cell membrane potential. Once the membrane potential exceeds a certain threshold, a spike is generated, and then the potential is set to a reset value. LIF [20] can capture the temporal information transmitted in SNN, which can be defined as:

$$\tau \frac{dV(T)}{dt} = -(V(t) - V_{reset}) + X(t), \qquad (1)$$

where $V(t)$ is the cell membrane potential at time t, $X(t)$ denotes the inputs at time t, τ is the membrane time constant, V_{reset} is the reset value after one spike is generated. The threshold of potential can be represented by V_{th}, the generation of one spiking at time t can be formatted as:

$$\mathrm{S}(t) = \begin{cases} 0, V(t) < V_{th} \\ 1, V(t) \geq V_{th} \end{cases}, \qquad (2)$$

where 1 is a spike and 0 means no operation. Generally, the τ in Eq. (1) is a constant. Based on the case analysis in [20], the Parametric Leaky Integrate-and-Fire (PLIF) spiking neuron model is proposed to adjust the τ during the training phase. To incorporate the expressiveness of the novel neuron type, PLIF is adopted herein as a fundamental structure of the SNN framework for dynamic vision emotion recognition. Following the strategy in [20], the surrogate gradient method is used to make backpropagation-based learning work.

2.3 SNN Framework

The neuron cells in the brain are connected by synapses. High biological plausibility requires the connections of neurons in SNN to mimic the true structure in the brain. However, biological scientists are still studying the connections of the brain, and some regional structures have been used to build neural networks. The convolutional layer's design is an imitation of the GCs part in the human brain while the pooling layer achieves a similar invariant effect of CCs in V1 and V4 [22]. Therefore, the convolutional layer and pooling layer are adopted to make up the SNN framework to complete dynamic vision emotion recognition.

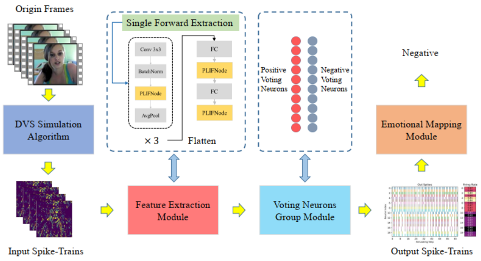


Figure 1. The framework of the emotion recognition

The aim of dynamic vision emotion recognition is to analyze emotional results given a series of data recording a specific scenario. The proposed SNN Framework consists of three parts, including a feature extraction module to extract informative features from input spike-trains, a voting neuron group module to give spike-trains of different emotion neuron groups, and an emotional mapping module to convert the spike-trains to final emotion polarity results. The overview of the framework including the dataset simulation process is illustrated in Fig. 1. The details of the DVS Simulation Algorithm have been introduced before and other modules will be presented below.

1) Feature Extraction Module: Feature Extraction Module (FEM) is utilized to extract informative features from input spike-trains. The original frames of videos are encoded into spike-trains. Different from the real data recorded by DVS, whose temporal resolution is high, the temporal resolution of spike-trains generated based on video frames is determined by the temporal resolution of videos. This schema of organizing data excludes the step of converting the asynchronous event stream into frames. Moreover, as mentioned before, the first frame is set to float-value which is more resembles the real application scenario. The component of FEM utilized herein is inspired by [20], where convolution, batch normalization, max-pooling, and PLIF neuron are adopted. Different from [20], the max-pooling is replaced by average pooling (AvgPool) based on the experimental results. Experimental results prove the superiority of average pooling on our dataset. This may be due to the tremendous information loss in spike-trains of max-pooling, as stated in [23].

2) Voting Neurons Group Module: To obtain the final emotion results, there need neurons to represent the corresponding emotion label. Suppose there are two emotion labels: positive emotions and negative emotions. In traditional ANNs, two different neurons are generally used directly to represent two different emotions. But in the human brain, the transmission of information is often carried out through a group of neurons. Two neuron populations composed of 10 neurons are applied to represent the final output: 10 positive voting neurons and 10 negative voting neurons. The informative features from FEM are divided into two groups of voting neurons. This module has no additional parameters, which only operated as reorganizing the spike-trains from another perspective. Thus, the final outputs of the voting module are spike-trains of 20 neurons and the length of each train is the simulating steps.

3) Emotional Mapping Module: Emotional Mapping Module is served as a classifier to conduct the final emotion recognition, specifically, to translate the output spike-trains into emotion labels. In SNN, fire rates of neurons during the simulation steps, denoted by N, are applied to represent the contribution to the corresponding target. One potential way is to directly define the desired spike trains for each emotion label. However, the definition is intricate and tedious [24] and the measurement techniques of two spike-trains are not as mature as a measurement of two float-value vectors. Thus, the average fire rate of 10 positive voting neurons is treated as the final output of emotion recognition. It is the same for negative voting neurons. If the average fire rate of positive voting neurons is larger than that of negative voting neurons, the input scene is positive, and vice versa. To observe in spike-trains, a higher fire rate means a dense distribution of spikes during the simulation period.

The average fire rate makes the measurement of output simple and the Mean Squared Error (MSE) is used to measure the average fire rates between the ground-truth emotion labels. The surrogate gradient method is applied to update the parameters in the framework by backpropagation.

3. Experiments

In this section, we firstly introduce a popular dataset for emotion recognition based on video segments and the simulation DVS based on this dataset will be presented. Then, the experimental setting is followed to detail the SNN settings. Finally, the results of experiments will be analyzed to verify the functionality of SNN based on the simulated dataset.

3.1 Dataset

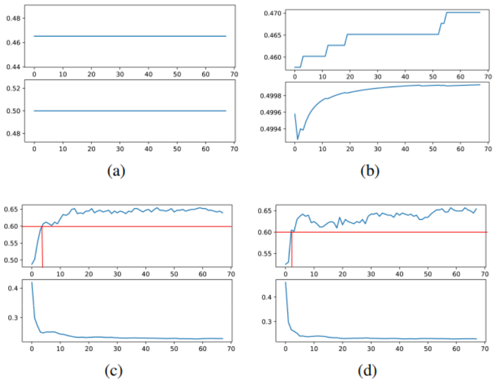
To conduct experiments with the previous model, a simulation dataset needs to be constructed first. Herein Carnegie Mellon University Multimodal Opinion Sentiment Intensity (CMU-MOSI) [21], a dataset recognized by the community and widely used, is chosen as the basis for the DVS simulation algorithm. The number of categories is mapped to two (positive and negative) based on the original label. Following the schedule mentioned before, a simulation DVS dataset can be generated. Note that the hyperparameter, $Sens$, will influence the simulation results. A larger $Sens$ will make the generated spike-trains sparser and a smaller $Sens$ will generate more spikes. Generally speaking, dense spike trains can achieve better accuracy, and sparse spike trains can potentially achieve lower energy consumption. $Sens$ is set to 0.001 to trade-off the accuracy and energy consumption.

3.2 Experimental setting

TABLE I Emotion recognition on simulated dataset

Algorithms	Explanation	Accuracy
PLIF-Fang [20]	network in [20]	61.69

ANN	counterpart of baseline SNN	65.17
baseline SNN	baseline structure mentioned with LIF	61.69
PLIF SNN	change the cell type of baseline	63.43
0.8SNN	change the voltage threshold of PLIF SNN	64.68
0.4SNN	change the voltage threshold of PLIF SNN	65.42
mpSNN	change to max-pooling of 0.4SNN	63.43
pureSNN	omit the first float-valued frame of 0.4SNN	65.17



(a) (b) (c) (d)

Figure 2. Curves of accuracy (above in each subfigure) and loss (bottom in each subfigure) on testing set: The x-axis represents different simulating steps, and the y-axis represents the testing accuracy and loss. (a) Before train with pure spike-trains as input. (b) During train with pure spike-trains as input. (c) After train with pure spike-trains as input. (d) After train with float-value as first frame.

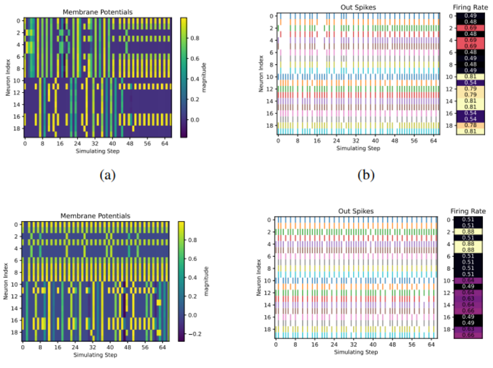
Experiments are implemented by SpikingJelly [25], a framework for SNN. The code runs on a Linux system with four Tesla V100 graphics cards. Initialization of the weights of synapses is completed by the default method of PyTorch with a fixed random seed [26]. To optimize the parameters, the stochastic gradient descent optimizer based on surrogate gradients implemented in SpikingJelly is utilized and the learning rate is 0.01. The batch size is set to 8 for all experiments. To find an appropriate time length of the input signal, we count the frame numbers of video segments and find that a large number of video segment samples are around 68. Thus, we set the time length to 68 herein. The input of neural networks is rescaled to 128×128. In order to make the comparison as fair as possible, an ANN basically parallel to the SNN structure is constructed, which is called the counterpart ANN, where the PLIF is changed to ReLU. The classifier for ANN is designed as a fully connected layer. We evaluate the performance of the SNN and its counterpart ANN on the simulation dataset.

3.3 Experimental Results

The performance of the testing set is summarized in Table I. The first column represents the compared methods or different hyper parameter setting algorithms. The second column is the explanation of the algorithms. The last column reported the accuracy of corresponding methods. Compared with the PLIF-Fang from [20], the counterpart ANN of baseline SNN gives a performance increase of 3.48%. It can be seen that the performance of PLIF-Fang and baseline SNN is the same, which can be explained by the influence of different network structures. Although LIF is applicable to the baseline SNN here, and its performance is theoretically worse than PLIF-Fang. But the pooling method is different: the max-pooling is utilized in PLIF-Fang while average pooling is adopted in baseline SNN.

The performance difference of these two pooling methods on the simulated dataset can be seen from the algorithms of 0.4SNN and mpSNN in Table I. The max-pooling will damage the performance of the model, which may be due to the loss of some local information compared with average pooling. Comparing baseline SNN to PLIF SNN, we can find that the performance of networks constructed with the PLIF neuron model is better (2.24% accuracy improvement), which is consistent with the conclusion in [20]. However, a different setting about the voltage threshold is presented based on our experiments on our dataset. The default voltage threshold is 1, it is thought unnecessary to adjust the voltage threshold in [20]. But in practice, we find that a relatively decreasing voltage threshold can obtain performance gain. It is shown that 0.4SNN achieves a better emotion recognition performance than its counterpart ANN. We argue that a relatively smaller voltage threshold of membrane potentials fires the neurons earlier, which causes a relative more training to reach a better performance. Finally, a pure spike-trains input setting is conducted to validate the effectiveness of the first float-value frame setting. It can be seen that compared with 0.4SNN, the pureSNN has a certain loss of performance.

To demonstrate what the training influence the SNNs' output at every simulating step (the total simulating herein is 68), experiments are conducted. The setting model using pure spike-trains as input is adopted to show the effect of training. The accuracy curve and loss curve on the testing set are shown in Fig. 2. In each subfigure, the accuracy curve is shown above, and the loss curve is shown below. It is shown in Fig. 2(a) that the SNNs give the same output for all simulating steps before training. This is due to the random initialization of parameters and no spike is fired in the classifier layer before training. During the early phase of training, some changes can be observed in Fig. 2(b) to suggest the updates of weights. After training is done, the loss in Fig. 2(c) decreases as the simulating step is larger at first, but then the loss curve starts to increase a little. A possible explanation is that the following spikes make the model out a result different ground truth. The fluctuation of the accuracy curve also illustrates the influence caused by following spikes after the simulating step achieving the lowest loss. In order to illustrate the superiority of the setting that using float-value in the first frame, we visualized the loss and accuracy curve during the training process in Fig. 2(d). It can be observed that compared to the pure spike-train input setting, using the float-value as the first frame in input can lead to faster loss drops at the beginning of training and make the accuracy exceed 60% earlier.



(a)

(b)

(c)

(d)

Figure 3. Examples to show the output spike potentials and output spike-trains: The x-axis represents different simulating steps, and the y-axis represents the membrane potential and output spike-trains in the voting neurons group module. Output spike potentials of correct sample (a) and wrong example (c). Output spike-trains of correct sample (b) and wrong sample (d).

TABLE II The number of operations and energy consumption of ANN and SNN on the generated dataset. # denotes the number of operations. M means million.

	#First layer	#Other Layers	Energy consumption per operation	Total consumption	Total consumption on the dataset
ANN	310.9M	47.9M	4.6pJ	1.65mJ	1.65mJ
SNN	9.1M	47.9M	0.9pJ	1.57mJ(N=30)	0.41mJ

To show the membrane potentials and spikes' pattern vividly, two examples are shown in Fig. 3. Note that there are ten neurons corresponding to positive and negative emotions and the output is defined as the average firing rate. For membrane potentials in Fig. 3(a), for every neuron on the y-axis, the rectangular bars in yellow represent the high potential, which is relatively easy to fire a spike. The output spike-trains corresponding to these membrane potentials are shown in Fig. 3(b). Neuron indexes 0 to 9 represent the negative emotion output and neuron indexes 10 to 19 represent the negative output. We can see that the spikes present two different patterns. For the first example with the positive emotion label shown in Fig. 3(b), from a visual point of view, the output spike-trains generated by the last 10 neurons are denser than the output spike-trains generated by the first 10 neurons. From the numerical analysis, the average spiking rate of the first ten neurons is 0.4925, which is smaller than that of the last ten neurons, which is 0.5075. Thus, the emotion is positive. For the second example with the negative emotion label shown in Fig. 3(d), The visual analysis is similar to the previous example. From the numerical analysis, the average spiking rate of the first ten neurons is 0.5060, which is larger than that of the last ten neurons, which is 0.4955. Thus, the emotion is negative, but the ground truth is positive.

To demonstrate the efficientness of the SNN, the energy consumption of SNN and counterpart ANN is analyzed theoretically. Note that the operations in SNN are mainly accumulation (ACC) while operations in ANN are Multiply-ACcumulation (MAC) [27]. It has been shown that a 32-bit floating-point MAC operation consumes 4.6 pJ while an ACC operation consumes 0.9 pJ in 45nm 0.9V chip [28]. The total related information is reported in Table II. As the structures of SNN and counterpart ANN are most the same, the number differences of MAC and ACC are caused by the input. The input frame's size is 128×128 and the filter kernel size is 3×3. For ANN with 68 channel inputs, the operation number of MAC is 310.9M. For SNN with 2 channel (positive events and negative events) input, the operation number of ACC is 9.1M. It should be noted that the operation of the first layer will be changed from ACC to MAC under the SNN with first frame float-values. The other part of the calculation of the same structure is 47.9M. The total consumption of ANN is 4.6× (47.9+310.9)=1.65mJ. For SNN, the first round is computed separately and the total consumption is (9.1 × 4.6)+9.1 × (N-1) × 0.9+47.9×N×0.9=(51.3×N+33.67)× 10^{-3} mJ. Consequently, it can be calculated the consumption of SNN is smaller than counterpart ANN when N<32. We can see from Fig. 2, the performance is almost stable when simulating step at around 30. A critical point is that when SNN is implemented by neuromorphic hardware [29, 30], the computation of SNN will exclusively happen when there is a spike. By counting the proportion of the number of spikes in the dataset to the total frame data, it is found that only 10.79% of positions occur spike events. In other words, the potential effectiveness advantage of SNN is larger than the above-mentioned. Thus, the total consumption of SNN with 68 simulating steps is about 0.41mJ. Compared with ANN's energy consumption, SNN's energy consumption is reduced by three quarters.

4. Conclusion

In this paper, we have proposed a simulated method to generate DVS-like data based on video segments and an SNN framework considering the real application scene to complete recognition. Inspired by the float input in ANN, the first frame of input to SNN is changed from spikes to float-value. The proposed SNN framework presents a feature extraction module for informative spike patterns from simulated input spike-trains and employs a voting neurons group module and emotion mapping module to convert output spike-trains to the final emotion labels. In addition, in our dataset, the theoretical energy consumption of SNN is only a quarter of that of ANN. An interesting future direction is to further explore the topology of other potential structures for SNN.

5. Acknowledgements

This work was supported by the Natural Science Foundation of Shandong Province (No. ZR2021QF145)

6. References

[1] S. Chen, A. Halimi, X. Ren, A. Mccarthy, and GS Buller. “Learning non-local spatial correlations to restore sparse 3d single-photon data,” IEEE Transactions on Image Processing, PP(99), 2019.

[2] G. Chen and X. Zeng. “Multi-modal emotion recognition by fusing correlation features of speech-visual,” IEEE Signal Processing Letters, 28:533–537, 2021.

[3] Y. Fu, L. Guo, L. Wang, Z. Liu, J. Liu, and J. Dang. “A sentiment similarity-oriented attention model with multi-task learning for text-based emotion recognition,” In International Conference on Multimedia Modeling, pages 278–289. Springer, 2021.

[4] K. Maher, Z. Huang, J. Song, X. Deng, Y. Lai, C. Ma, H. Wang, Y. Liu, and H. Wang. "E-ffective: A visual analytic system for exploring the emotion and effectiveness of inspirational speeches," IEEE Transactions on Visualization and Computer Graphics, 28(1):508–517, 2021.

[5] L. Sun, B. Liu, J. Tao, and Z. Lian. "Multimodal cross-and self-attention network for speech emotion recognition," In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 4275–4279. IEEE, 2021.

[6] B. Wang, G. Dong, Y. Zhao, R. Li, Q. Cao, and Y. Chao. "Non-uniform attention network for multi-modal sentiment analysis," In International Conference on Multimedia Modeling, pages 612–623. Springer, 2022.

[7] Y. Zhang, G. Zhao, Y. Shu, Y. Ge, D. Zhang, Y. Liu, and X. Sun. "Cped: A chinese positive emotion database for emotion elicitation and analysis," IEEE Transactions on Affective Computing, 2021.

[8] J. Chen, D. Jiang, Y. Zhang, and P. Zhang. "Emotion recognition from spatiotemporal eeg representations with hybrid convolutional recurrent neural networks via wearable multi-channel headset," Computer Communications, 154:58–65, 2020.

[9] L. Wang and K. Yoon. "Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021.

[10] T. Elsken, J. Metzen, and F. Hutter. "Neural architecture search: A survey. The Journal of Machine Learning Research," 20(1):1997–2017, 2019.

[11] W. Maass. "Networks of spiking neurons: the third generation of neural network models," Neural networks, 10(9):1659–1671, 1997.

[12] E. Benssassi and J. Ye. "Investigating multisensory integration in emotion recognition through bio-inspired computational models," IEEE Transactions on Affective Computing, 2021.

[13] C. Buscicchio, P. Górecki, and L. Caponetti. "Speech emotion recognition using spiking neural networks," In International Symposium on Methodologies for Intelligent Systems, pages 38–46. Springer, 2006.

[14] R. Lotfidereshgi and P. Gournay. "Biologically inspired speech emotion recognition," In 2017 IEEE international conference on acoustics, speech and signal processing (ICASSP), pages 5135–5139. IEEE, 2017.

[15] Y. Luo, Q. Fu, J. Xie, Y. Qin, G. Wu, J. Liu, F. Jiang, Y. Cao, and X. Ding. "Eeg-based emotion classification using spiking neural networks," IEEE Access, 8:46007–46016, 2020.

[16] E. Benssassi and J. Ye. "Speech emotion recognition with early visual cross-modal enhancement using spiking neural networks," In 2019 International Joint Conference on Neural Networks (IJCNN), pages 1–8. IEEE, 2019.

[17] C. Posch, D. Matolin, and R. Wohlgenannt. "A qvga 143 db dynamic range frame-free pwm image sensor with lossless pixel-level video compression and time-domain cds," IEEE Journal of Solid-State Circuits, 46(1):259–275, 2010.

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

[19] J. Wu, Y. Chua, and H. Li. "A biologically plausible speech recognition framework based on spiking neural networks," In 2018 International Joint Conference on Neural Networks (IJCNN), 2018.

[20] 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, pages 2661–2671, 2021.

[21] A. Zadeh, R. Zellers, E. Pincus, and L. Morency. "Mosi: multimodal corpus of sentiment intensity and subjectivity analysis in online opinion videos," arXiv preprint arXiv:1606.06259, 2016

[22] Q. Xu, Y. Qi, H. Yu, J. Shen, H. Tang, G. Pan, et al. "Csnn: An augmented spiking based framework with perceptron-inception," In IJCAI, pages 1646–1652, 2018.

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

[24] Q. Liu, G. Pan, H. Ruan, D. Xing, and H. Tang. "Unsupervised aer object recognition based on multiscale spatio-temporal features and spiking neurons," IEEE Transactions on Neural Networks and Learning Systems, PP(99):1–12, 2020.

[25] W. Fang, Y. Chen, J. Ding, D. Chen, Z. Yu, H. Zhou, Y. Tian, and other contributors. “Spikingjelly. https://github.com/fangwei123456/spikingjelly”, 2020.

[26] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. “Pytorch: An imperative style, high-performance deep learning library,” Advances in neural information processing systems, 32:8026–8037, 2019.

[27] B. Chakraborty, X. She, and S. Mukhopadhyay. “A fully spiking hybrid neural network for energy-efficient object detection,” arXiv preprint arXiv:2104.10719, 2021.

[28] M. Horowitz. “1.1 computing’s energy problem (and what we can do about it),” In 2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC), pages 10–14. IEEE, 2014.

[29] 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, 77:43–51, 2017.

[30] J. Pei, L. Deng, S. Song, M. Zhao, Y. Zhang, S. Wu, G. Wang, Z. Zou, Z. Wu, W. He, et al. “Towards artificial general intelligence with hybrid tianjic chip architecture,” Nature, 572(7767):106–111, 2019.