Using Spiking Neural Networks to Classify the ASL Alphabet

*Note: ASL is short for American Sign Language

Mehul Gupta
Computer Science, Senior
Tickle College of Engineering
Knoxville, TN
mgupta13@vols.utk.edu

Brian Stewart
Computer Science, Senior
Tickle College of Engineering
Knoxville, TN
bstewa43@vols.utk.edu

Channing Tan
Electrical Engineering, Senior
Tickle College of Engineering
Knoxville, TN
ctan9@vols.utk.edu

***Abstract*—This project explores the use of spiking neural networks (SNNs) for recognizing American Sign Language (ASL) hand gestures from neuromorphic event-based camera data. Traditional frame-based vision technologies process redundant spatial information, whereas neuromorphic cameras capture asynchronous pixel-level changes, making them well-suited for energy-efficient and temporally precise gesture recognition. In short, neuromorphic event-based cameras capture changes in pixel brightness which is useful for capturing and focusing on only hand movement for gesture classification. The key objective of this project is to determine the effectiveness of an SNN implementation for learning and classifying ASL signs from the event-based data provided by the ASL-DVS dataset.**

To handle and preprocess the neuromorphic data, event streams were organized into structured spike-based frames using temporal binning and denoising techniques. A convolutional SNN (CSNN) architecture was implemented using leaky integrate-and-fire (LIF) neurons, enabling the model to process spatiotemporal patterns across multiple time steps. The network was trained on labeled ASL samples representing 24 static hand signs (excluding dynamic letters) from 5 different subjects and optimized using spike-based loss functions.

The model achieved high training accuracy, reaching approximately 96.7% after 7 epochs, indicating strong learning capability on the training data. However, test performance remained low, suggesting challenges with generalization, likely due to limited data diversity, overfitting, or suboptimal hyperparameters. These results highlight both the potential and current limitations of SNNs for event-based gesture recognition tasks as well as areas for possible future improvements to the model.

I. Introduction and Motivation

The overarching goal of this project is to develop a system capable of recognizing American Sign Language (ASL) hand gestures using neuromorphic event-based camera data and a spiking neural network (SNN). Unlike traditional image classification approaches that rely on static frames, this project focuses on processing temporally rich, asynchronous data to better capture the dynamics of hand movements and shapes. The system aims to classify 24 ASL letters (excluding dynamic gestures such as ”j” and ”z”) by learning spatiotemporal patterns in event streams. To achieve this goal, we analyzed the ASL-DVS dataset, which currently stands as the largest event-based dataset for ASL gestures [1].

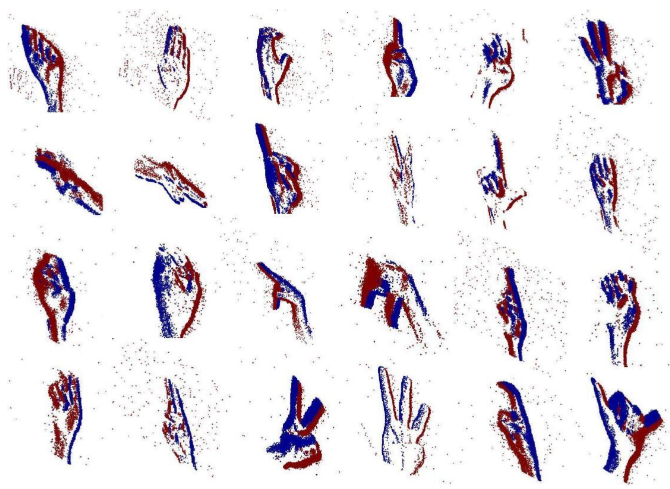

Fig. 1. Visualization of the ASL-DVS dataset [1].

The primary research question addressed in this work is: *Can a spiking neural network effectively learn and generalize ASL gesture patterns from neuromorphic event-based data?* More specifically, the project investigates whether the temporal processing capabilities of SNNs provide an advantage when working with sparse, time-dependent inputs compared to conventional neural network approaches.

This topic was selected for both practical and technical reasons. From a personal perspective, ASL recognition presents an engaging and meaningful application, as it could serve as a tool to assist individuals who are not familiar with sign language. From a technical standpoint, the use of neuromorphic data and SNNs introduces a novel and less commonly explored paradigm in machine learning. Neuromorphic cameras capture changes in pixel intensity over time rather than full images, making the data inherently temporal and sparse. This aligns well with the design of SNNs, where neurons accumulate input over time and emit spikes once a threshold is reached. The biologically inspired mechanism of integrating information through time makes SNNs particularly suitable for this type of data.

Additionally, this project provided an opportunity to explore emerging technologies and gain hands-on experience with event-driven data processing and bio-inspired learning models, both of which are becoming increasingly relevant in energy-efficient and real-time AI systems.

II. Related Work

The capabilities of SNNs was formalized by Maass [2] who had demonstrated their theoretically greater computational power over traditional neural network approaches of the time, such as models based on McCulloch Pitts neurons. Unlike conventional deep learning approaches, such as convolutional neural networks (CNNs), SNNs process discrete spikes over a given time period, making them well suited for processing temporal data, further enabling energy efficient computation on neuromorphic hardware [3].

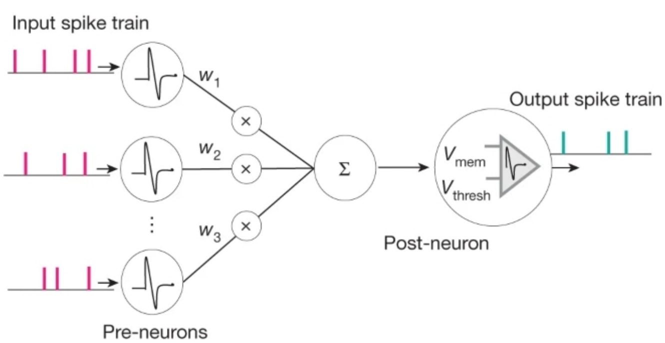

Fig. 2. Visualization of a SNN, with input pre-neural spikes producing output post-neural spikes. [3].

The utility of SNNs has been made more apparent with the development of neuromorphic hardware, such as the dynamic vision sensor (DVS), which was pioneered by Lichtsteiner et al. [4]. The DVS they had introduced was capable of recording a 128x128 asynchronous pixel grid, where each pixel would act independently to detect changes in local relative intensity, which enabled the sensor to produce spike events. The sensor produces these sparse, low-latency event streams that naturally align with the temporal processing capabilities of SNNs. Gallego et al. [5] provide a comprehensive overview of the field of event based vision, discussing DVS designs, their applications in fields such as robotics, and the methods that would be used to process the event-based data generated by DVS, establishing the broader context in which this work sits.

A closely related work is that of Amir et al. [6], which discusses the development of an energy efficient end-to-end gesture recognition system implemented through a DVS and a CNN configuration on IBM's TrueNorth neurosynaptic processor. Their work demonstrated the feasibility of real-time gesture recognition utilizing event-based neuromorphic hardware, as they had achieved a strong accuracy of 96.5% on their dataset consisting of 11 separate hand gestures. Our work aimed to expand upon this by targeting the 24 ASL letters present in the ASL-DVS [1] dataset, which would introduce more challenges of inter-class similarities.

Sign language recognition has been widely studied through a more traditional computer vision lens. Such is seen in the work of Molchanov et al. [7] which showcased the application of 3D convolutional networks to dynamic hand gesture recognition utilizing continuous RGB and infrared data. Their model presented a strong accuracy, only slightly below human accuracy, yet it could not leverage the sparse, energy-efficient data that would be produced by a DVS camera. Furthermore, Pigou et al. [8] explored recurrent and convolutional methods for continuous gesture recognition to demonstrate the ineffectiveness of a simple temporal feature pooling strategy for accounting the temporal aspects of video. While successful in their approach, their work also examines video, rather than leveraging event-based data to exploit its inherent temporal nature.

This work has been expanded upon in more recent years as others such as Ceolini et al. [9] have begun to utilize DVS cameras and event-based data for gesture recognition tasks. In their work they had demonstrated hand gesture recognition that utilizes a sensor fusion framework, which relies on electromyography (EMG) signals from muscles, as well as visual information. This multi-sensor approach held a high degree of accuracy and robustness, though it also came with a high computational cost. As such, they were able to utilize the low computational cost of neuromorphic technologies, namely DVS cameras and neuromorphic processors, to counteract the high cost. Their work showcased the robustness that could come from fusing sensor modalities.

Compared to these prior works, this project specifically targets ASL alphabet recognition from purely event-based data. To do this, we focused on implementing a convolutional spiking neural network (CSNN) that was trained with surrogate gradients, without a reliance on the fusion of other data modalities.

III. Method/Approach/Experimental Setup

This project implements a spiking neural network (SNN)-based pipeline for classifying American Sign Language (ASL) gestures from neuromorphic event data. The system is built off of the PyTorch [10] and SnnTorch [11] libraries, which provide support for biologically inspired neuron models such as leaky integrate-and-fire (LIF) neurons, as well as surrogate gradient methods for training. Additionally, the Tonic library [12] is used for preprocessing event-based data, including denoising and conversion into frame-based spike representations.

The ASL-DVS dataset [1] was utilized for all training and testing purposes within this work. This dataset consists of 100,800 samples, with each sample lasting for approximately 100ms, which were reordered from 5 subjects utilizing a iniLabs DAVIS240c NVS set up in an environment with low environmental noise as well as a constant illumination. Each class contains 4,200 samples to ensure a balanced class distribution. The dataset contains each of the letters in the ASL

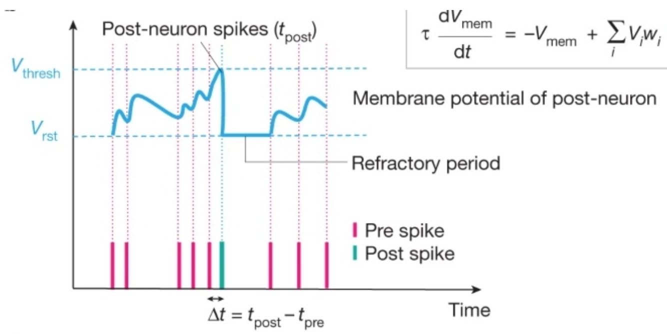

Fig. 3. A representation of the dynamics of LIF neurons. The membrane potential spikes and leaks over time before reaching a given threshold and firing, after which the neuron enters a refractory period. [3].

alphabet with the exclusion of ”j” and ”z”, as these gestures require dynamic motion, which makes them fundamentally different from the rest of the alphabet which can be represented by static hand positions. For our purposes, we used a subject based split, with subjects 1 through 4 being used for training, while subject 5 was utilized for testing. The purpose of this was to both avoid data leakage and evaluate the model's ability to generalize on an entirely unseen individual, which would provide a more realistic evaluation on the model's performance.

Raw input data consists of event streams captured by a neuromorphic camera, where each event encodes spatial coordinates, timestamp, and polarity. Polarity consists of either on/off, representing if the pixel increased or decreased in brightness. An example representation of the neuromorphic data is shown below.

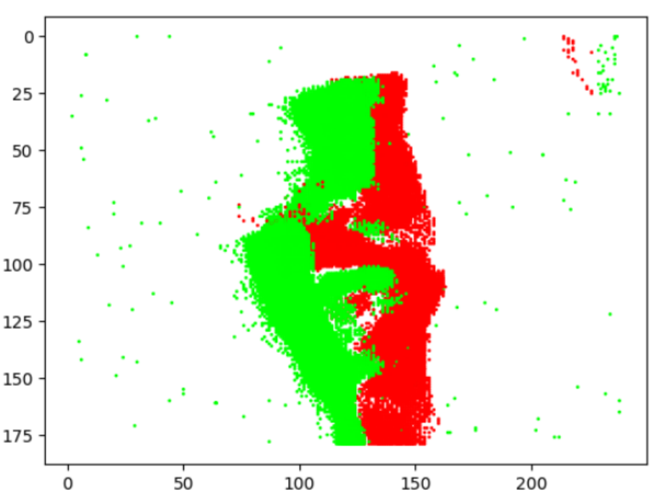

Fig. 4. Example of neuromorphic camera data. Green pixels are ones that increased in brightness while red indicates a decrease in brightness.

These events are first converted into a structured format and passed through a preprocessing pipeline that applies temporal denoising and bins events into a fixed number of time steps. This results in a tensor representation of shape (time x channels x height x width) suitable for input into the SNN model.

The architecture of the model is illustrated in Figure 5. The network consists of 2 convolutional layers (3x3 kernels) followed by LIF neuron layers to introduce temporal dynamics. A max-pooling layer reduces spatial dimensionality, and a fully connected layer maps extracted features to 24 output classes corresponding to ASL letters. At each time step, the network processes input spikes sequentially, and neuron membrane potentials accumulate information over time before generating output spikes.

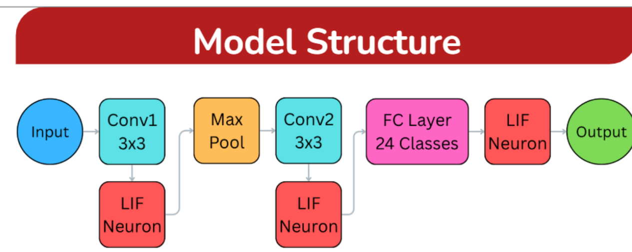

Fig. 5. ASL SNN Model Structure.

Experimentally, the model is trained using a supervised learning approach with a spike-based loss function that encourages correct firing behavior for target classes. Training is conducted over multiple epochs using the Adam optimizer, with performance evaluated using classification accuracy. The dataset is split into training and testing sets with a fixed number of samples drawn from each file (40 per class) to control dataset size and training time. The primary aspect being investigated is the model's ability to learn meaningful spatiotemporal patterns from event-based input and generalize them to unseen data.

The source code for this project is publicly available at: https://github.com/BrianStew/CS420FInalProject.

IV. Results

The performance of the proposed spiking neural network (SNN) was evaluated using training loss, training accuracy, and a confusion matrix on the test set. These results provide insight into both the learning behavior of the model and its ability to generalize to unseen data. Figures depicting the performance of the proposed model is shown below.

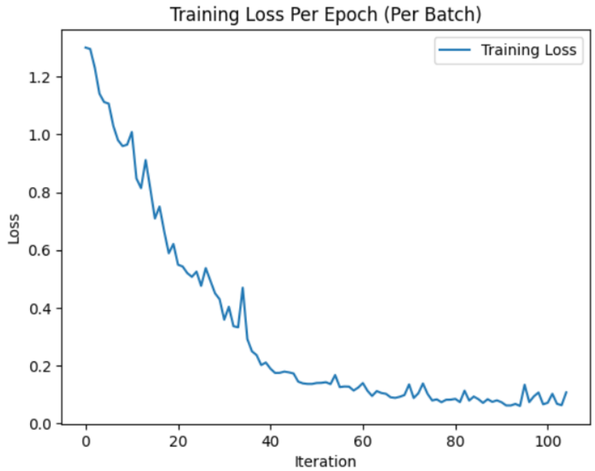

Fig. 6. CSNN Training Loss.

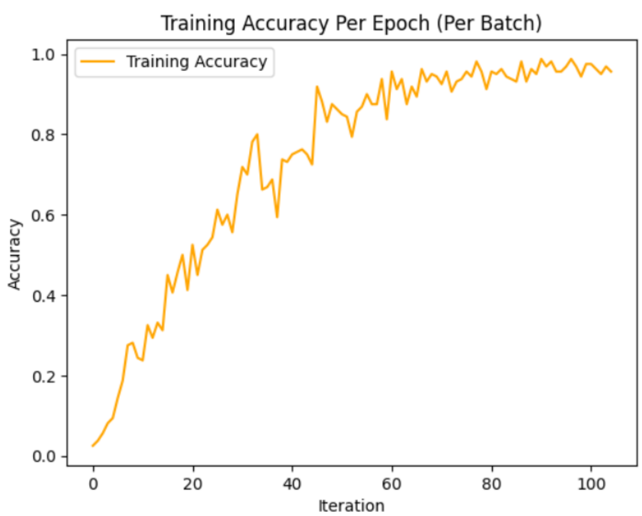

Fig. 7. CSNN Training Accuracy.

The training accuracy curve shows a steady and significant improvement over time, increasing from near 0% to approximately 95-97% by the end of training. This indicates that the model successfully learned to classify the training data and was able to extract meaningful spatiotemporal features from the neuromorphic input. Similarly, the training loss decreases consistently throughout training, dropping from above 1.2 to below 0.1. The smooth downward trend in loss suggests stable optimization and effective learning using the spike-based loss function.

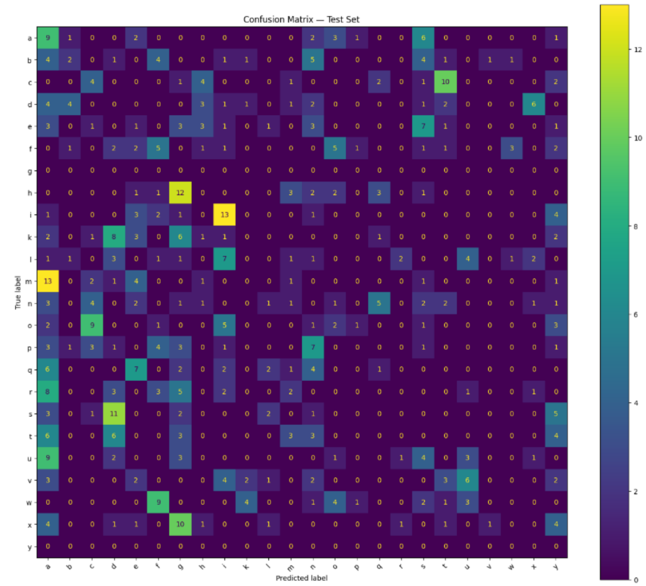

Fig. 8. Confusion Matrix showing the results from the test set (subject 5) when evaluated with the trained CSNN. Rows represent true labels, columns represent predicted labels.

However, the confusion matrix for the test set reveals a stark contrast in performance. Predictions are widely distributed across incorrect classes, with very few strong diagonal elements, indicating correct classifications. This is further confirmed by the overall test accuracy of approximately 4.56%, which is significantly lower than the training accuracy. The confusion matrix also does reveal some logical misclassifications, as there are similarities between some of the classes. For example, the letter ”a” is frequently misclassified as ”m”, which would make sense as the signs for those letters are fairly similar. It is also interesting that the model was capable of correctly classifying ”i” specifically, while struggling with the rest of the classes. If the results were purely random, it would be unlikely to have such a high success rate on a specific class.

This discrepancy between training and testing performance suggests that the model is overfitting to the training data and failing to generalize. Several factors may contribute to this outcome. First, the dataset size is limited due to restricting the number of samples per file, which may prevent the model from learning robust, generalizable features. This was a choice made to reduce training time to a workable time-frame given our hardware limitations. Second, variability between subjects as well as recording conditions could introduce differences between training and test data distributions. Furthermore, the SNN architecture and hyper-parameters, while effective for learning training patterns, may not be sufficiently regularized for generalization. It is also worth noting that some incorrect classifications could be due to very similar patterns and gestures for different letters. For example, in American Sign Language (ASL), the letters A and E are often considered very similar because they both involve a closed fist, but they are distinguished by the position of the thumb. Other pairs that are frequently confused due to similar hand-shapes and orientations include:

- G and Q: Both are made with the index finger and thumb, but ”G” points sideways, while ”Q” points downward
- M and N: Both involve folding fingers over the thumb, but ”M” covers three fingers (including the ring finger), while ”N” covers only two
- S and T: Both are closed fists, but ”S” has the thumb across the front of the knuckles, while ”T” has the thumb tucked between the index and middle fingers

Since several pairs of letters are very close visually, this could be difficult for the model to distinguish, especially with our restricted dataset, which could lead to many incorrect classifications as well.

Overall, while the model demonstrates strong learning capability on the training set, the results highlight a key limitation in its ability to generalize, emphasizing the need for improved data diversity, model tuning, or regularization techniques in future work.

V. Discussion, Conclusion, and Future Work

The results of this project demonstrate that spiking neural networks (SNNs) are capable of learning meaningful spatio-temporal representations from neuromorphic event-based data. The model achieved high training accuracy (approximately 96.7%), indicating that the architecture-combining convolutional layers with leaky integrate-and-fire (LIF) neurons-was effective at capturing patterns within the training set. The steady decrease in training loss further supports that the network was able to converge and learn the underlying structure of the data.

However, the extremely low test accuracy (approximately 4.56%) highlights a major limitation in the model's ability to generalize. This suggests that the network overfit to the training data and failed to learn features that transfer well to unseen samples. Potential causes include limited dataset size, insufficient variability in training samples, suboptimal hyper-parameters, and a lack of regularization within the model. Additionally, inconsistencies in the data or preprocessing pipeline may have contributed to the gap between training and testing performance.

A significant constraint in this project was hardware availability. Training SNNs on event-based data is computationally intensive due to the temporal dimension and sequential processing across time steps. One team member had access to a desktop with an NVIDIA RTX 3070 GPU, while other experimentation had to be conducted using cloud-based environments such as Google Colab utilizing the A100 GPUs, which proved to be slow and restrictive. These limitations made it difficult to iterate upon the model, tune hyper-parameters, increase the number of training samples per file, or train for additional epochs. As a result, optimally tuning and exploring the capabilities of the model was not as feasible as we had originally desired.

For future work, there are several improvements that could be made. Firstly, increasing the dataset size and diversity would likely improve generalization. Additionally, having access to more powerful and consistent hardware would enable more thorough experimentation, such as hyper-parameter tuning, deeper model architectures, as well as longer training durations. Furthermore, the incorporation of regularization techniques such as dropout, data augmentation, or normalization could help reduce the model's tendency to overfit to the training data. Given more time, we could aim to explore alternative SNN architectures, different encoding schemes for event data, or hybrid models that combine traditional neural networks with spiking components may also yield better performance. Finally, extending the system to handle dynamic ASL gestures (such as ”j” and ”z”) would provide a more complete and practical application.

Overall, this project demonstrates both the promise and challenges of applying SNNs to neuromorphic data, highlighting the importance of data quality, model design, and computational resources in achieving strong real-world performance.

References

[1] Y. Bi, A. Chadha, A. Abbas, , E. Bourtsoulatze, and Y. Andreopoulos, “Graph-based object classification for neuromorphic vision sensing,” in *2019 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2019.

[2] W. Maass, “Networks of spiking neurons: The third generation of neural network models,” *Neural Networks*, vol. 10, no. 9, pp. 1659–1671, 1997. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608097000117

[3] K. Roy, A. Jaiswal, and P. Panda, “Towards spike-based machine intelligence with neuromorphic computing,” *Nature*, vol. 575, no. 7784, pp. 607–617, Nov 2019. [Online]. Available: https://doi.org/10.1038/s41586-019-1677-2

[4] P. Lichtsteiner, C. Posch, and T. Delbruck, “A 128$\times$ 128 120 db 15 μs latency asynchronous temporal contrast vision sensor,” *IEEE Journal of Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, Feb 2008.

[5] G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, “Event-based vision: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 1, pp. 154–180, Jan 2022.

[6] A. Amir, B. Taba, D. Berg, T. Melano, J. McKinstry, C. Di Nolfo, T. Nayak, A. Andreopoulos, G. Garreau, M. Mendoza, J. Kusnitz, M. Debole, S. Esser, T. Delbruck, M. Flickner, and D. Modha, “A low power, fully event-based gesture recognition system,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[7] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz, “Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[8] L. Pigou, A. van den Oord, S. Dieleman, M. Van Herreweghe, and J. Dambre, “Beyond temporal pooling: Recurrence and temporal convolutions for gesture recognition in video,” *International Journal of Computer Vision*, vol. 126, no. 2, pp. 430–439, Apr 2018. [Online]. Available: https://doi.org/10.1007/s11263-016-0957-7

[9] E. Ceolini, C. Frenkel, S. B. Shrestha, G. Taverni, L. Khacef, M. Payvand, and E. Donati, “Hand-gesture recognition based on emg and event-based camera sensor fusion: A benchmark in neuromorphic computing,” *Frontiers in Neuroscience*, vol. Volume 14 - 2020, 2020. [Online]. Available: https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2020.00637

[10] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019,

pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[11] J. K. Eshraghian, M. Ward, E. O. Neftci, X. Wang, G. Lenz, G. Dwivedi, M. Bennamoun, D. S. Jeong, and W. D. Lu, "Training spiking neural networks using lessons from deep learning," *Proceedings of the IEEE*, vol. 111, no. 9, pp. 1016–1054, 2023.

[12] G. Lenz, K. Chaney, S. B. Shrestha, O. Oubari, S. Picaud, and G. Zarrella, "Tonic: event-based datasets and transformations." Jul. 2021, Documentation available under https://tonic.readthedocs.io. [Online]. Available: https://doi.org/10.5281/zenodo.5079802