electronics

MDPI

Article

Sign Language Gesture Recognition and Classification Based on Event Camera with Spiking Neural Networks

Xuena Chen [1], Li Su [1,*], Jinxiu Zhao [1], Keni Qiu [1], Na Jiang [1] and Guang Zhai [2]

[1] School of Information Engineering, Capital Normal University, Beijing 100048, China
[2] School of Aerospace Engineering, Beijing Institute of Technology, Beijing 100081, China
* Correspondence: li.su@cnu.edu.cn

Abstract: Sign language recognition has been utilized in human–machine interactions, improving the lives of people with speech impairments or who rely on nonverbal instructions. Thanks to its higher temporal resolution, less visual redundancy information and lower energy consumption, the use of an event camera with a new dynamic vision sensor (DVS) shows promise with regard to sign language recognition with robot perception and intelligent control. Although previous work has focused on event camera-based, simple gesture datasets, such as DVS128Gesture, event camera gesture datasets inspired by sign language are critical, which poses a great impediment to the development of event camera-based sign language recognition. An effective method to extract spatio-temporal features from event data is significantly desired. Firstly, the event-based sign language gesture datasets are proposed and the data have two sources: traditional sign language videos to event stream (DVS_Sign_v2e) and DAVIS346 (DVS_Sign). In the present dataset, data are divided into five classification, verbs, quantifiers, position, things and people, adapting to actual scenarios where robots provide instruction or assistance. Sign language classification is demonstrated in spike neuron networks with a spatio-temporal back-propagation training method, leading to the best recognition accuracy of 77%. This work paves the way for the combination of event camera-based sign language gesture recognition and robotic perception for the future intelligent systems.

Keywords: event camera; spiking neural network; DVS-sign language; sign language recognition; intelligent system

Citation: Chen, X.; Su, L.; Zhao, J.; Qiu, K.; Jiang, N.; Zhai, G. Sign Language Gesture Recognition and Classification Based on Event Camera with Spiking Neural Networks. *Electronics* **2023**, *12*, 786. https://doi.org/10.3390/electronics12040786

Academic Editors: Zhan Li, Zhang Chen, Yiyong Sun and Maria Evelina Fantacci

Received: 30 November 2022
Revised: 21 January 2023
Accepted: 26 January 2023
Published: 4 February 2023

1. Introduction

Sign language recognition can help people with speech impairments break through communication barriers in social life, and also it can be used in human–machine interaction to enrich nonverbal instructions. Sign language recognition has attracted much attention in the field of computer vision, with the goal to accurately recognize the movement of gestures and understand the meaning of sign language for communicators or machines. The recognition of sign language actions has huge application in robot perception [1], improving the lives of people with speech impairments [2] and enriching nonverbal information transfer. However, since sign language movements are generally too fast, the use of traditional frame cameras introduces great challenges, such as blur and overlap and computational complexity.

Inspired by the biological vision mechanism, a new type of dynamic vision sensor (DVS) event camera has become popular, in which each pixel independently detects the brightness changes and generates asynchronous event streams. It has significant technical and application advantages compared with traditional cameras: a high time resolution (microsecond level), low delay (μs level), low power consumption (10 mW) and high dynamic range (120–143 dB) [3]. It opens up the possibility of developing a promising method for gesture recognition with robot perception and intelligent control. DVS128Gesture [4], which combines the event camera and gesture recognition, does not consider the abundant

meanings of sign language. A few studies [5,6] combined an event camera with sign language gesture recognition, but there are few publicly available gesture recognition datasets and benchmarks.

For the training algorithm suitable for the combination of event cameras and sign language gesture recognition, we chose the spiking neural network (SNN), which is suitable for asynchronous and event-driven tasks. It has the ability to analyze and process event stream data generated by event cameras. Because of the relatively rich traditional video format sign language dataset, the proposed dataset can be obtained by using the video to event (v2e) [7] method. On the other hand, we created an event-based sign language dataset (called DVS_Sign) using the event camera DAVIS346. DVS_Sign contains a total of 600 training sign language vocabulary videos, with five classifications of sign language classified by the part of speech.

The contributions of this work are as follows: (1) Event camera-based sign language gesture datasets are proposed, named DVS_Sign and DVS_Sign_v2e. Considering human–machine interaction, we chose some common sign language gestures for the proposed dataset, which were divided into five classifications—verbs, quantifiers, position, things and people. These were adapted to actual scenarios where robots need instructions and assistance. (2) Sign language gesture recognition and classification are demonstrated in a lightweight SNN with spatio-temporal back propagation (STBP) method, taking advantage of both the event camera and SNN, achieving up to 77% accuracy.

2. Related Work

We investigated some sign language datasets, including the United States (ASL-LEX), Boston (ASLLVD), American Sign Language (ASL), Argentinian (LSA64 [8]), and China. Some countries have their own specific languages. We considered the issues of different countries and human–machine interactions to design our dataset. Even though a few studies proposed an event camera-based American sign language dataset, they cannot have not been open-sourced to estimate the performance [5]. For SL-Animals-DVS, which uses gestures to imitate animals, there is no specific sign language vocabulary [6]. ASL-DVS [9] contains 24 classes corresponding to 24 letters from ASL. DVS128Gesture is a public event camera-based gesture dataset, which only contains 11 common simple gestures, such as left hand wave, right hand clockwise, arm drums. In this work, our own collection of event sign language datasets using the event camera DAVIS346, as shown in Figure 1. Figure 1a shows the asynchronous event flow data collected when posing in front of the event camera. Figure 1b shows two consecutive grayscale images of the event sign language visualization from an event camera. The experiment benefits from the asynchronous nature and ease of processing of event camera data.

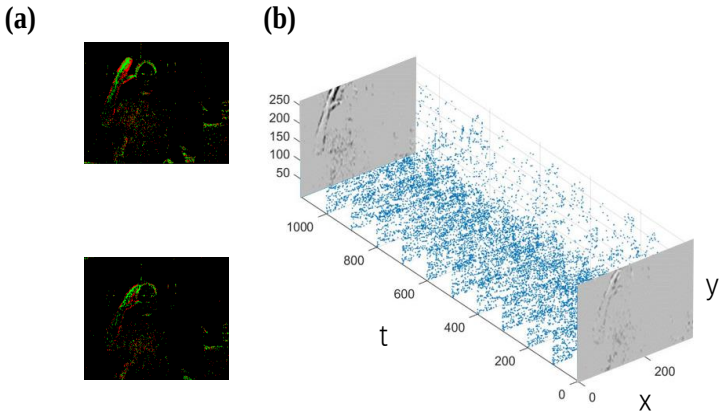

Figure 1. The visualization of DAVIS346 [10] collecting event sign language dataset. (**a**): DVS sensor generates asynchronous event stream when a sign language is posing in front of it; (**b**): continuous raw events between two consecutive grayscale images from an event camera.

Refs. [11,12] developed efficient feature representation methods for event streams, and some studies have developed event-based gesture recognition systems. Lee et al. [13] first developed an event-based gesture recognition system with DVS and proposed a primitive event called the leaky integrate-and-fire (LIF) neuron approach. A recent work [9] proposed dynamic vision sensors with graph-based spatio-temporal features, which contributed greatly to action recognition using event-based cameras. Although gesture recognition has seen great progress and achieved great success in various applications, there are still some important factors that reduce the robustness of recognition systems.

In order to better extract the features of sign language actions and perceive fine-grained temporal and spatial features from event stream data, there are many event-based action recognition and classification methods [4,14–17]. Point cloud-based [18] and graph-based methods [14,17] treat event data as point clouds or graph nodes. However, converting raw event data into this format leads to the fine-grained temporal and spatial information contained in the event data being discarded. SNN-based methods [19–21] use spike neural networks [22] to process input event streams asynchronously; however, they are difficult to train because of the lack of efficient back-propagation [23] algorithms. Existing CNN-based methods [24,25] transform asynchronous event data into fixed-rate frame-like representations and feed them into standard deep neural networks [26]. The time resolution according to the event frame leads to a loss of information in other spatial or temporal dimensions. In summary, traditional event-based motion detection methods [27,28] are not suitable for our sign language gesture recognition, which requires fine-grained spatio-temporal [29] feature detection from event data.

Although traditional sign language gesture recognition based on an RGB camera has achieved high accuracy in an ideal environment, the cost of consumption includes a large number of training samples and complex calculations. Due to the data from the RGB camera, high-accuracy sign language recognition cannot be realized in a fast or dark environment. This is also the reason that we investigate this method based on the event camera and reduce the cost through SNN to overcome the issues in sign language recognition. SNN is known as a third-generation neural network that attempts to more closely match the function of biological brains; specifically, the membrane potential accumulates input over time and sends out a spike when a set threshold is crossed. Furthermore, in the event camera, each pixel independently detects brightness changes and generates asynchronous event streams, matching the event-based nature of SNNs. Therefore, sign language gesture recognition with event cameras is considered to be combined with SNN.

3. Method

3.1. Introduction of Event Data

The event stream consists of pixel array, the trigger time and polarity (signal of brightness change). Triggered events are expressed as:

$$e = (x, y, t, \rho)^T \tag{1}$$

Event e represents the image of the event camera at $(x, y)^T$. The event triggered by the brightness change in the pixel at time t, $\rho(p, c)$ is a truncation function.

$$\rho(p, c) = \begin{cases} +1, & (p \geq c) \\ -1, & (p \leq -c) \end{cases} \tag{2}$$

c is the excitation threshold of the event point, p is the brightness change value when the brightness increment is more than c, the positive polarity event point is excited when the brightness increment is less than $-c$, and the negative polarity event point is excited when the absolute value of the brightness increment is c. When it is less than c and more than $-c$, the event camera has no output.

3.2. Sign Language Dataset

Due to the lack of sign language datasets based on event cameras, we chose the v2e method to convert traditional RGB video sign language videos into event stream data.We used the LSA64 dataset, which contains 64 commonly used words in daily life. Verbs and nouns are included. We chose some common sign language gestures for the proposed dataset considering human–machine interaction. Additionally, to better validate our training method, we created the DVS_Sign dataset, the event-based sign language dataset, using event cameras. We used the event camera DAVIS346, which can output event stream and intensity information at the same time using the event camera to collect the same sign language video as the LSA64 dataset, and package each sign language video in the same folder. Since the collected sign language dataset is in aedat format, we converted aedat into the csv data format required by the network. This ensured a more accurate comparison of the impact of datasets on training accuracy. We provided a video-to-event method to obtain data from different existing sign language datasets from RGB video, as a technical solution to sign language dataset conversion.

There are three volunteers that participate in the recording of the indoor scene sign language dataset. In order to avoid each action of the volunteers being too similar, we shuffled the recording order to ensure the accuracy of recording each sign language action. Volunteers were asked to sit in front of a DAVIS346 and posed for each sign language word. In addition, in order to ensure the accuracy of the sign language movements, we asked the volunteers to wear simple clothes without too much decoration and try to ensure that they worn short tops during the recording process. We required volunteers to have no additional clothing decorations during the recording process, so as to avoid excessive event data noise from interfering with the experimental results. Considering the practical applicability of our work, in the process of selecting sign language datasets, we chose some commonly used imperative words, such as some positions, the prescriptive word, such as "right, shut down, accept" and so on.Taking into account the actual application scenarios of this experiment and the actual needs of the people with speech impairment, we chose these vocabularies such as "call, away, help" in our datasets. The sign language meanings represented by the specific serial numbers are shown in Table 1.

Table 1. The division of the part of our sign language datasets and the corresponding display of categories and serial numbers.

Word Category	Name	ID
verb	help	00
	accept	05
	away	01
	take	11
	shut down	08
	thanks	03
	call	10
quantifier	1	02
	5	06
things	water	13
	music	04
position	right	07
	left	09
people	mom	12
	dad	14

Figure 2 shows the visual display of some words in DVS_Sign language gesture dataset, which could be applied to scenarios of practical application of human–machine or robotic instructions. For examples, "one water, thanks" and "call mom" will be recognized by machine or robot, so that they can carry out some simple instructions, which is also in line with the original intention of our work. In addition, we will consider adding more gestures and optimizing the network structure to better perform sign language gesture recognition in more complex scenes.

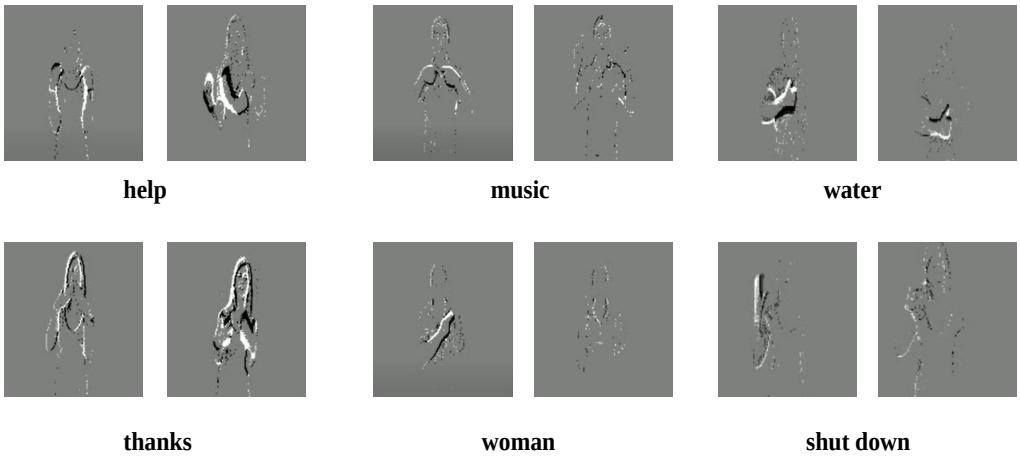

Figure 2. Part of the visual display of DVS_Sign language gesture dataset, each vocabulary example selects 2 grayscaled event video frames.

3.3. Network Structure of SNN

Only considering the spatial domain (as supervised via back-propagation) or the temporal domain (as unsupervised with temporal plasticity) leads to a performance bottleneck due to the existing action classification training algorithms [30,31]. Therefore, building a learning framework that fully utilizes the spatio-temporal domain (STD) is the fundamental requirement for high-performance SNNs. In our method, capturing spatio-temporal features in the event stream is very important to improve recognition accuracy. LIF is the most widely used model to describe neuron dynamics in SNNs, which can be simply given by

$$\tau \frac{\mathrm{d}u(t)}{\mathrm{d}t} = -u(t) + I(t) \tag{3}$$

$$u(t) = u(t_{i-1} e^{\frac{t_{i-1}-t}{\tau}}) + I(t) \tag{4}$$

τ is a time constant, $u(t)$ represents the neuron membrane potential at time t, and $I(t)$ denotes the pre-synaptic input, determined by pre-neuronal activity or external injection and synaptic weight. When the membrane potential u exceeds a given threshold V_{th}, the neuron triggers a spike and resets its potential to u_{reset}. As shown in Figure 3, the incoming event frame which are collected from DAVIS346 and event converted by v2e, are forward-propagated through the SNN network, and a pooling operation is performed before each layer of convolution. For the gradient descent of iterative representation, in SNN we pass the chain rule of layer-by-layer reverse error propagation—in others words, iterative SNN based on LIF. At the same time, self-feedback injection at each neuron node generates a non-volatile integral in TD as shown in Equation (4). It used $u(t)$ to approximate the neuronal potential based on the last spiking moment t_{i-1} and the pre-synaptic input $I(t)$; the membrane potential decays exponentially until the neuron receives a pre-synaptic input, and once the neuron spikes, a new round of updates begins. In other words, the spatial accumulation $I(t)$ and the leaked event memory $u(t_{i-1})$ determine the state of

the neuron. After our encoding layer, there are two flattened layers at the end to predict the output of the classification results of sign language.

$$\begin{cases} V_n^t = H_{t-1}^n + \frac{1}{\tau}(I_{t-1}^n - (H_{t-1}^t - V_{reset})) \\ S_t^n = \Theta(V_t^n - V_{threshold}) \\ H_t^n = V_t^n \cdot (1 - S_t^n) \end{cases} \tag{5}$$

In Equation (5), n and t represent the number of layers and time steps of the network, S represents the peak tensor with a binary value, and I represents the input of the previous layer. The membrane potential continues to decay until the next new pulse is fired until a new input is received. So, the neuron state is jointly determined by the pre-synaptic potential $I'(t)$ and the leaked membrane potential $u(t_{i-1})$ at the previous moment. Θ denotes the Heaviside step function [32], H is the reset process after spiking.

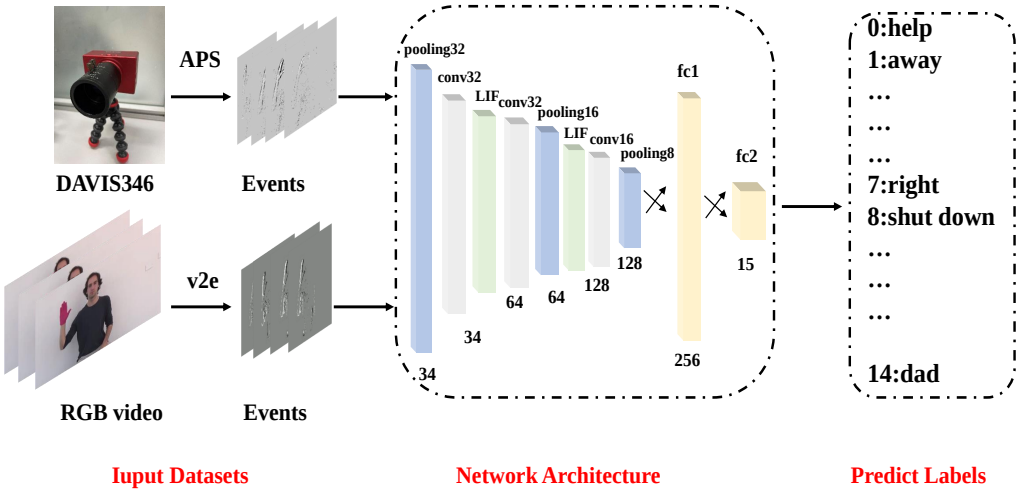

Figure 3. The process of sign language gesture recognition and classification. The two datasets in the figure do not enter the network at the same time, so the drawing is for a more intuitive display.

The back-propagation efficiency of the traditional neural network greatly benefits from the iterative representation of gradient descent, so the LIF-based iterative SNN is used in our training. Algorithm 1 describes the iterative update of a single LIF neuron at time steps $t+1$ and with $n+1$ layers, where the f function represents SpikeAct, which will output a spike and perform back-propagation. If the input membrane potential $u^{t+1,n+1}$ exceeds Threshold V_{th}, a spike is generated, and then the newly generated membrane potential and spike output are returned to the next neuron, and so on.

Algorithm 1 State update for an explicitly iterative LIF neuron at time step $t+1$ in the $n+1$ layer

Require: previous potential $u^{t,n+1}$ and spike output $o^{t,n+1}$, current spike input $o^{t+1,n}$, and weight vector W
Ensure: next potential $u^{t+1,n+1}$ and spike output $o^{t+1,n+1}$
1: **function** STATE_UPDATE(W,$u^{t,n+1}$,$o^{t,n+1}$,$o^{t+1,n}$)
2: $u^{t+1,n+1} = k_{\tau 1} u^{t,n+1}(1\text{-}o^{t,n+1})+W\ o^{t+1,n}$
3: $o^{t+1,n+1} = \mathrm{f}(u^{t+1,n+1}\text{-}V_{th})$
4: **return** $u^{t+1,n+1}$,$o^{t+1,n+1}$
5: **end function**

4. Experiments

In this section, we introduce our experimental method in details, including the data collection and data prepossessing, the selection of the loss function in our training process,

and the impact of the adjustment of each parameter in the STBP training method on the experimental results.

4.1. Dataset Prepossessing

Due to the lack of event camera-based sign language datasets, our experiments were initially based on traditional RGB sign language videos converted into event stream data through the v2e method. The following is the detailed process of the conversion data prepossessing:

Steps A–B in Figure 4a represent the color-to-brightness conversion. A color RGB video is transformed into M luminance frames, where each frame is associated with a timestamp. Then, synthetic slow motion is introduced: the luma frames are optionally interpolated using the Super-SloMo video inter-polation network [33] to increase the temporal resolution of the input video. Super-SloMo predicts the bi-directional optical flow vectors from successive luminance frames. This is used to linearly interpolate new frames at arbitrary positions between the two input frames. The next step is a linear to logarithmic mapping [34]: a standard digital video usually represents intensity linearly, but DVS pixels detect changes in logarithmic intensity. Step C represents the event generation model; we assume that the pixel has a memorized brightness value and the new low-pass filtered brightness value. The model was used to generate a signed integer quantity to represent positive ON or negative OFF events from the change of memorized brightness value minus filtered brightness value, and the luminance value stored in multiple DVS events is updated to a signed integer multiple of the threshold. (b) means that we use DAVIS346 to collect sign language datasets, and the size of the collected video is 346 × 260. Secondly, we denoised the event stream data, cleaned up unnecessary event coordinates, and then cropped the video size to 128 × 128, so as to enter the network training, and finally the event frame dataset of our sign language dataset was obtained.

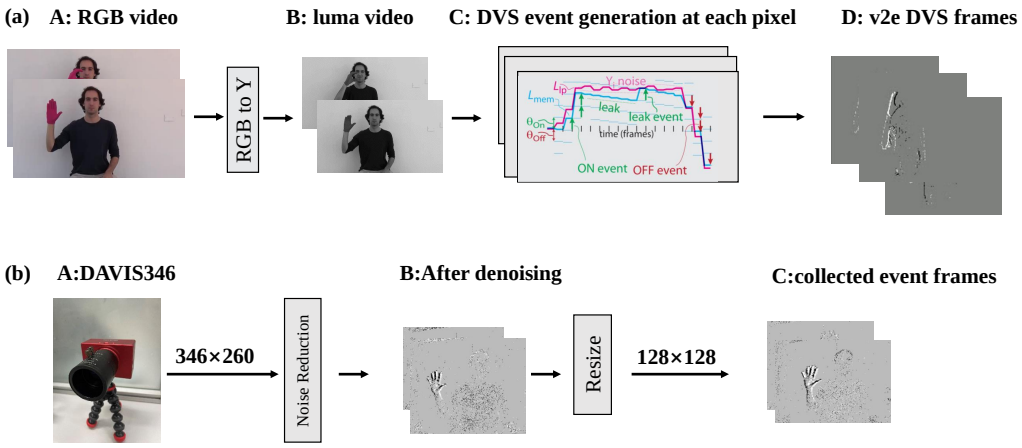

Figure 4. Two datasets processing methods. (**a**): use the v2e method to convert RGB video to event stream data; (**b**): use DAVIS346 to collect sign language event video streams for processing.

In addition, we also used the DAVIS346 to collect the dataset for comparison with our v2e dataset. Considering the practical applicability of the experimental results, we conducted a lot of work in selecting the dataset not only included some commonly used sign language gestures. In addition to the robot imperative vocabulary, we also chose some vocabularies of machine instructions, because they are very important for the daily needs of people with speech impairments and robots, which are lack of hearing function and nonverbal commands. We also included a large vocabulary in the scope of experimental data collection. In the future, as the experiment progresses, we will consider adding more sign language vocabulary.

4.2. Implementation Details

We tested the STBP training method of our SNN model on two datasets, including the selection of the loss function in Equation (6) and the adjustment of the V_{th} parameter, and we also describe the back-propagation training process and setting of initialization of each parameter in detail.

(1) Loss function

During the experiment, we tried to replace the loss function with the commonly used cross-entropy loss [35] and the softmax function [36] for multi-classification [37]. After applying the above-mentioned cross-entropy loss function and softmax loss function to our experiments, the experimental accuracy of the mean square error is the highest, which is due to the fact that in STBP under the training framework, the loss function of the mean square error ensures that the error of the output and label of each sample during the gradient return process is the smallest.

$$L = \frac{1}{2S}\sum_{s=1}^{S}||y_s - \frac{1}{T}\sum_{t=1}^{T}o_s^{t,N}||_2^2 \tag{6}$$

In Equation (6), we show our loss function $\boldsymbol{L}$ and minimize the mean squared error (MSE) of all samples over a given time window $\boldsymbol{T}$, where $\boldsymbol{y_s}$ and $\boldsymbol{o_s}$ represent the label vector of the $\boldsymbol{s_{th}}$ training sample and the neuron output vector of the last layer $\boldsymbol{N}$, respectively.

(2) Back-propagation Training

Figure 5 shows the error transmission in the SD (spatial domain) and TD (timing-dependent temporal domain) on a single neuron, vertical and horizontal paths, respectively.

At the single neuron level, Figure 5, the propagation is decomposed into a vertical path for SD and a horizontal path for TD, each neuron accumulates a weighted error signal from the upper layer and iteratively updates parameters in different layers. In TD, the neuron state is iteratively expanded in the temporal direction that enables chain rule propagation. At this point, we clearly understand that the complete gradient descent process is obtained during the training process. On one hand, each neuron accumulates weighted error signals from the upper layers in SD, on the other hand, each neuron also expands the state space iteratively based on the chain rule from the received propagated errors in self-feedback dynamics.

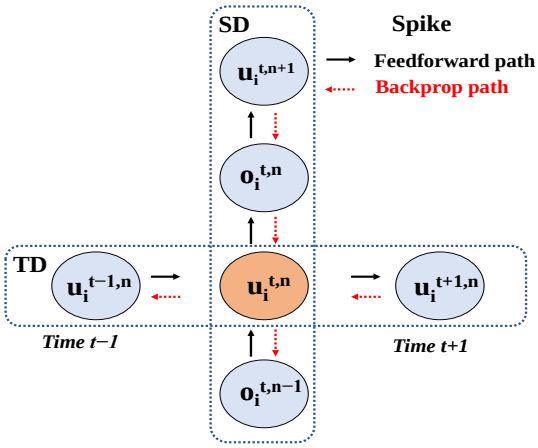

Figure 5. At the single-neuron level, the vertical path and horizontal path represent the error propagation in the SD and TD, respectively.

(3) Rate coding [38,39]

We use frequency coding in our network.Within the simulation time T, each cycle determines whether to send a spiking with the probability of the spiking firing $rate \cdot dt$ (dt is in seconds) and finally forms a frequency code.

Assuming that spiking-counting frequency serves as the basis for encoding in the nervous system, the frequency with neurons fire contains all the information. The spiking counting frequency can be measured only by calculating the number of spikings in the time interval; in other words, calculating the time average. The frequency v of neuron spiking firing can be understood as the ratio of the average number of spikings $n_{sp}(T)$ observed in a specific time interval T divided by time T:

$$v = \frac{n_{sp}(T)}{T} \tag{7}$$

The encoding time window T is experimentally based on actual data and depends on the type of neuron and the form of stimulation. In our experiments, this is 40 ms or 60 ms. According to the results of our experiments, T is determined by the number of the event in the time interval. If the time interval T is too large or too small, it is not conducive to the accumulation of spikes and an appropriate time window T is selected according to different experimental data and event data. The frequency encoding process based on spiking counting is shown in the figure below. Figure 6a describes that the frequency v of a neuron is the average of the number of spikings released at a given time T. Figure 6b shows the output frequency v as a function of the total input current I_o in the gain function. As the stimulus intensity increases, the neuron spiking firing frequency v gradually increases, and for a larger input current I_o, the firing frequency gradually approaches the maximum value v_{max}.

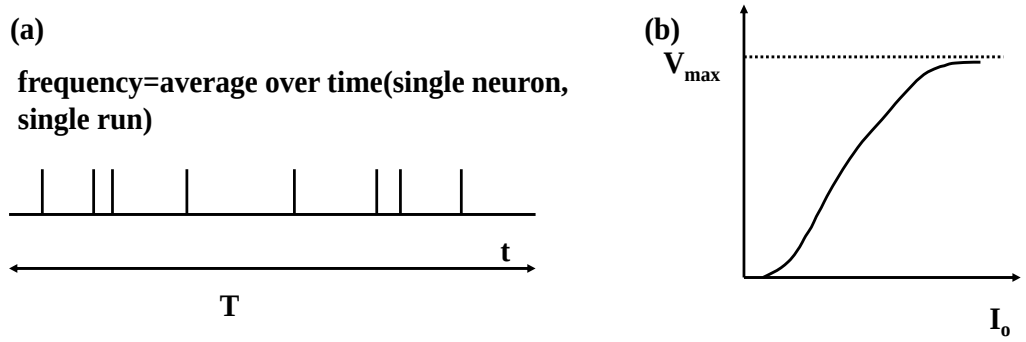

Figure 6. (**a**): Spiking firing frequency defined by event averaging. (**b**): Schematic diagram of the gain function.

4.3. Experimental Details

All experiments in this work are performed on our own collection of 15 commonly used sign language datasets and on the same 15 v2e-transformed event stream sign language datasets based on LSA64. The input space dimension of our proposed network model is 128 × 128, which does not need to be cropped on the v2e datasets, because it has been adjusted when converting the original video into an event stream during the data prepossessing. However, in the dataset collected by ourselves above, we used DAVIS346, the size of the captured video frames are all 346 × 260, so when the data are fed into our network model, they are cropped so that it can adapt to the training dimension of our network model. In addition, because the event data collected by DAVIS346 are too large and has too much noise, we denoised the collected event stream video to reduce unnecessary event coordinate data and prevent interference in the process of network feature extraction.

We use the PyTorch framework [40] to implement all the methods used in this work. Our model is optimized by SGD [41] using standard settings. We adjusted the learning rate

during training according to Equation (8), where the initial learning rate is set to 1×10^{-4}. We set the batch size to 20 and trained the model for 200 epochs. We divided the test set because during the training process we found that too many training samples entered the network and the test accuracy of the training model was not very good, so the dataset was divided into two parts—training and testing.

$$lr_{new} = lr \cdot (0.1^{(epoch \div 60)}) \tag{8}$$

5. Experimental Results

The spatio-temporal back-propagation method is trained and tested on two datasets, and the experimental results are shown in Table 2. The sign language types of both datasets are identical, except for the data prepossessing method. We split the training set and the testing set 4:1.There are a total of 15 kinds of sign language videos, each with 40 videos in the training set and 10 videos in the validation set. So, the total training set has 600 and the test set has 150 and the validation set has 100. We trained it on the TITAN server by adjusting batch_size and the epochs.

Table 2. Comparison of the v2e dataset with the DAVIS346 acquisition dataset and other event-based and traditional video dataset-based action classification models. DVS_Sign_v2e is a dataset that converts traditional LSA sign language videos into event streams through the v2e method. DVS_Sign is the dataset collected by DAVIS346. Acc1 represents the accuracy of the first part of the test set; Acc2 represents the accuracy of the second part of the test set; Acc represents the accuracy of the entire test set.

Method	Dataset	Input	Backbone	Acc1 (%)	Acc2 (%)	Acc (%)
Ye et al. [42]	ASL	RGB	CNN	-	-	69.20
Zhang et al. [43]	EgoGesture	RGB	VGG16+LSTM	-	-	68.90
Xu et al. [24]	ASL-DVS	event	GIN	-	-	51.4
Monti et al. [44]	ASL-DVS	event	MoNet	-	-	86.7
Martinez et al. [18]	DVS-Lip	event	ResNet-18	55.60	75.46	65.51
Liu et al. [45]	DVS-Lip	event	ResNet-101	58.36	79.17	68.74
Ours	DVS_Sign_v2e	event	SNN+STBP	79.00	76.00	77.00
Ours	DVS_Sign	event	SNN+STBP	71.00	70.00	68.00

We compare our data results with several related actions classify methods as comparative experiments, including (1) event-based gesture action classification methods [18,24,44]; (2) traditional image-video-based action classification methods [42,43].

Compared with a traditional RGB camera, an event camera can accurately capture the motion information of sign language gesture and get rid of the weak light conditions and high speed [9,39]. DVS-Lip is applied in traditional neural networks, but it maybe fails to take advantage of the data characteristics of the event camera without utilizing SNN [24]. It can also be seen that there are some gaps compared to the ASL-DVS [8,13]. Since the simple gesture is for one letter in the ASL-DVS dataset, it has a higher accuracy for the recognition of each letter gesture, but it may result in a lower accuracy because it needs to "spell" the word to express the instructions. Compared with our sign language gesture recognition, it is easier for the network to extract feature information. From the tabular data above, we found that our training method significantly outperforms some existing event-based video-based action recognition classification methods on DVS_Sign_v2e datasets. This shows that our network model and dataset are effective, and feature information can be extracted from sign language event data. In addition to comparing it with other event-based action classifications, we can also see that there is also a gap in the test results before the two datasets. The test results of the dataset collected by DAVIS346 are worse than the test results converted from v2e. It may be because the light is related to clockwise

and counterclockwise gestures, changes in the environment, and changes in the distance during the process of data collection. There is another reason the DAVIS346 output event dimension size is 346 × 260, and the input size received by our network model is 128 × 128, before we crop the video frame size. In the process, some event point coordinates and timestamps that are used for feature extraction are lost.

Comparing the experimental results of the DVS_Sign_v2e dataset with event-based action recognition, we have made some progress in terms of accuracy, and because our network is relatively lightweight, the training time is also advantageous. Two hundred epochs only takes 15 hours. Our network has fewer layers than other networks. However, from the dataset, we collected DVS_Sign language, and found that the results were not so good. The first reason may be that there is a loss of event stream data when the collected event camera data are converted from aedat format to csv format; the second is that there is a loss when the captured event stream video of 346 × 260 size is cropped to 128 × 128. The third reason is that there are differences between our data due to the external environment when we collect data, such as the brightness of the light, the distance between the sign language movement and the event camera, the clockwise and counterclockwise direction of the sign language movement may affect the experimental results. This also shows that our training method STBP, the robustness of it is not strong enough. In the face of some situations where the data are not so clean, there has been a certain degree of loss, which is what we need to improve.

During the experiments, we found that the initialization of various parameters was very important to the experimental results, such as the threshold and weight and other parameters, which were crucial to the firing spike activity of the entire network. As shown in Table 3, we need to ensure the timely response of presynaptic stimulation. At the same time, avoiding excessive spike firing reduces the selectivity of neurons. Among them, dt, lr, step and threshold parameters have a great influence on the experimental results. After many experiments, we continued to adjust the method to find the best value suitable for different datasets. Among them, dt represents the time interval of taking an event frame for the dt time interval of the entire sign language dataset video, and step represents the number of internal loops during the training process. Among them, the learning rate and V_{th} parameters are the optimal solutions obtained from continuous adjustment after multiple training sessions and can be adjusted according to your own data set and data size. Among them, V_{th} represents the threshold value set in the process of generating spikes, and this value will determine the number of spikes generated.

Table 3. Comparison of parameters corresponding to the training results.

Dataset	Step	dt	V_{th}	lr	Acc (%)
DVS_Sign	50	40	0.4	4×10^{-4}	65.00
DVS_Sign	60	30	0.2	1×10^{-4}	68.00
DVS_v2e	70	50	0.2	4×10^{-3}	72.00
DVS_v2e	80	40	0.3	1×10^{-3}	77.00

6. Discussion

In this work, we consider the characteristics of low power consumption and high temporal resolution of event camera, which makes it suitable for sign language gesture recognition with robot perception and human machine interaction. The present DVS_Sign language gesture dataset is proposed and the recognition is demonstrated in SNN with STBP. Firstly, to better help people with speech impairments or nonverbal robot instructions, the dataset can be expanded and enriched in the future. Secondly, the data formats that our network model can accept are limited. If multiple data formats including the event data and traditional RGB video can both be received directly, the networks could be added data pre-processing module and improved in the efficiency for practical applications.

7. Conclusions

This paper introduces a method for sign language gesture recognition and classification based on an event camera, through the use of an event-based sign language gesture dataset (DVS_Sign and DVS_Sign_v2e), which is demonstrated in the SNN with the STBP method. The present dataset is divided into five classifications: verbs, quantifiers, position, things and people. These adapt to actual scenarios where robots provide instructions or assistances. Additionally, we trained on both datasets and the best result was 77% accuracy on the DVS_Sign_v2e dataset, which verified the feasibility and validity of this method. However, this work did not achieve highest accuracy, which can be further studied by modified the network structure and expanding the datasets to enrich the accuracy and robustness of the algorithm. In the future, we will consider taking more scenes of this sign language gesture recognition method into the field of robot perception. In addition to its low power consumption, the high-speed feature of the event camera is also significant for sign language gesture recognition and robotic perception in future intelligent systems.

Author Contributions: Conceptualization, L.S. and K.Q.; methodology, L.S., X.C. and G.Z.; software, X.C. and J.Z.; data collection and curation, X.C.; validation, X.C. and J.Z.; data curation and analysis, J.Z. and L.S.; visualization, X.C.; Writing—original draft, X.C. and J.Z.; Writing—review and editing, L.S., K.Q., N.J. and J.Z.; data analysis, K.Q., N.J. and G.Z.; supervision, K.Q. and L.S.; funding acquisition, L.S. and N.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by R&D Program of Beijing Municipal Education Commission (KM202110028010) and the National Natural Science Foundation of China (62002247) and State Administration of Science, Technology and Industry for National Defence, PRC (HTKJ2020KL502013).

Data Availability Statement: The data presented in this study are available on request from corresponding authors. In addition, our dataset has been open source, and our DVS-Sign datasets can be obtained at github: https://github.com/najie1314/DVS/tree/main (accessed on 30 November 2022) In addition, the DVS-Sign-v2e data set is at https://github.com /najie1314/DVS/tree/master (accessed on 30 November 2022) is also available.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Nihal, R.A.; Broti, N.M.; Deowan, S.A.; Rahman, S. Design and development of a humanoid robot for sign language interpretation. *SN Comput. Sci.* **2021**, *2*, 220. [CrossRef]
2. Fellinger, J. *Public Health of Deaf People*; Gallaudet University Press: Washington, DC, USA, 2015; pp. 111–130. [CrossRef]
3. Su, L.; Yang, F.; Wang, X.-Y.; Guo, C.-D.; Tong, L.-L.; Hu, Q. A survey of robot perception and control based on event camera. *Acta Autom. Sin.* **2022**, *48*, 1869–1889. [CrossRef]
4. Amir, A.; Taba, B.; Berg, D.; Melano, T.; McKinstry, J.; Di Nolfo, C.; Modha, D. A low power, fully event-based gesture recognition system. In Proceedings of the IEEE Conference on Computer Vision and Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7243–7252. [CrossRef]
5. Wang, Y.; Zhang, X.; Wang, Y.; Wang, H.; Huang, C.; Shen, Y. Event-Based American Sign Language Recognition Using Dynamic Vision Sensor. In *Wireless Algorithms, Systems, and Applications. WASA 2021. Lecture Notes in Computer Science*; Liu, Z., Wu, F., Das, S.K., Eds.; Springer: Berlin/Heidelberg, Germany, 2021; Volume 12939. [CrossRef]
6. Vasudevan, A.; Negri, P.; Di Ielsi, C.; Linares-Barranco, B.; Serrano-Gotarredona, T. SL-Animals-DVS: Event-driven sign language animals dataset. *Pattern Anal. Appl.* **2022** *25*, 505–520. [CrossRef]
7. Hu, Y.; Liu, S.C.; Delbruck, T. v2e: From video frames to realistic DVS events. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 1312–1321. [CrossRef]
8. Ronchetti, F.; Quiroga, F.; Estrebou, C.A.; Lanzarini, L.C.; Rosete, A. LSA64: An Argentinian sign language dataset. In Proceedings of the XXII Congreso Argentino de Ciencias de la Computación (CACIC), San Luis, Argentina, 3–7 October 2016.
9. Bi, Y.; Chadha, A.; Abbas, A.; Bourtsoulatze, E.; Andreopoulos, Y. Graph-based spatio-temporal feature learning for neuromorphic vision sensing. *IEEE Trans. Image Process.* **2020**, *29*, 9084–9098. [CrossRef] [PubMed]
10. Tedaldi, D.; Gallego, G.; Mueggler, E.; Scaramuzza, D. Feature detection and tracking with the dynamic and active-pixel vision sensor (DAVIS). In Proceedings of the 2016 Second International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP), Krakow, Poland, 13–15 June 2016; pp. 1–7. [CrossRef]
11. Xiao, R.; Tang, H.; Ma, Y.; Yan, R.; Orchard, G. An event-driven categorization model for AER image sensors using multispike encoding and learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *31*, 3649–3657. [CrossRef]

12. Lagorce, X.; Orchard, G.; Galluppi, F.; Shi, B.E.; Benosman, R.B. HOTS: A hierarchy of event-based time-surfaces for pattern recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1346–1359. [CrossRef] [PubMed]
13. Lee, J.H.; Delbruck, T.; Pfeiffer, M.; Park, P.K.; Shin, C.W.; Ryu, H.; Kang, B.C. Real-time gesture interface based on event-driven processing from stereo silicon retinas. *IEEE Trans. Neural Netw. Learn. Syst.* **2014**, *25*, 2250–2263. [CrossRef]
14. Ceolini, E.; Frenkel, C.; Shrestha, S.B.; Taverni, G.; Khacef, L.; Payvand, M.; Donati, E. Hand-gesture recognition based on EMG and event-based camera sensor fusion: A benchmark in neuromorphic computing. *Front. Neurosci.* **2020**, *14*, 637. [CrossRef]
15. Shrestha, S.B.; Orchard, G. Slayer: Spike layer error reassignment in time. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 31. [CrossRef]
16. Wang, Y.; Du, B.; Shen, Y.; Wu, K.; Zhao, G.; Sun, J. EV-gait: Event-based robust gait recognition using dynamic vision sensors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 6358–6367. [CrossRef]
17. Wang, Y.; Zhang, X.; Shen, Y.; Du, B.; Zhao, G.; Cui, L.; Wen, H. Event-stream representation for human gaits identification using deep neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 3436–3449. [CrossRef]
18. Martinez, B.; Ma, P.; Petridis, S.; Pantic, M. Lipreading using temporal convolutional networks. In Proceedings of the ICASSP IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 6319–6323. [CrossRef]
19. Zhang, J.; Dong, B.; Zhang, H.; Ding, J.; Heide, F.; Yin, B.; Yang, X. Spiking Transformers for Event-Based Single Object Tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 8801–8810. [CrossRef]
20. Cordone, L.; Miramond, B.; Thierion, P. Object Detection with Spiking Neural Networks on Automotive Event Data. *arXiv* **2022**, arXiv:2205.04339.
21. Zhu, L.; Wang, X.; Chang, Y.; Li, J.; Huang, T.; Tian, Y. Event-based Video Reconstruction via Potential-assisted Spiking Neural Network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 3594–3604. [CrossRef]
22. Liu, C.; Shen, W.; Zhang, L.; Du, Y.; Yuan, Z. Spike neural network learning algorithm based on an evolutionary membrane algorithm. *IEEE Access* **2021**, *9*, 17071–17082. [CrossRef]
23. Lillicrap, T.P.; Santoro, A.; Marris, L.; Akerman, C.J.; Hinton, G. Backpropagation and the brain. *Nat. Rev. Neurosci.* **2021**, *21*, 335–346. [CrossRef] [PubMed]
24. Xu, K.; Hu, W.; Leskovec, J.; Jegelka, S. How powerful are graph neural networks? *arXiv* **2018**, arXiv:1810.00826.
25. Sejuti, Z.A.; Islam, M.S. An efficient method to classify brain tumor using CNN and SVM. In Proceedings of the 2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), Haka, Bangladesh, 5–7 January 2021; pp. 644–648. [CrossRef]
26. Fang, W.; Yu, Z.; Chen, Y.; Huang, T.; Masquelier, T.; Tian, Y. Deep residual learning in spiking neural networks. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 21056–21069. [CrossRef]
27. Lu, Y.; Naganawa, M.; Toyonaga, T.; Gallezot, J.D.; Fontaine, K.; Ren, S.; Carson, R.E. Data-driven motion detection and event-by-event correction for brain PET: Comparison with Vicra. *J. Nucl. Med.* **2020**, *61*, 1397–1403. [CrossRef]
28. Vasco, V.; Glover, A.; Mueggler, E.; Scaramuzza, D.; Natale, L.; Bartolozzi, C. Independent motion detection with event-driven cameras. In Proceedings of the 2017 18th International Conference on Advanced Robotics (ICAR), Hong Kong, China, 10–12 July 2017; pp. 530–536. [CrossRef]
29. Wu, Y.; Deng, L.; Li, G.; Zhu, J.; Shi, L. Spatio-temporal back-propagation for training high-performance spiking neural networks. *Front. Neurosci.* **2018**, *12*, 331. [CrossRef]
30. Aarrestad, T.; van Beekveld, M.; Bona, M.; Boveia, A.; Caron, S.; Davies, J.; Zhang, Z. The dark machines anomaly score challenge: Benchmark data and model-independent event classification for the large hadron collider. *SciPost Phys.* **2022**, *12*, 043. [CrossRef]
31. Blance, A.; Spannowsky, M. Unsupervised event classification with graphs on classical and photonic quantum computers. *J. High Energy Phys.* **2021**, *2021*, 170. [CrossRef]
32. Kyurkchiev, N.; Markov, S. On the Hausdorff distance between the Heaviside step function and Verhulst logistic function. *J. Math. Chem.* **2016**, *54*, 109–119. [CrossRef]
33. Jiang, H.; Sun, D.; Jampani, V.; Yang, M.H.; Learned-Miller, E.; Kautz, J. Super slomo: High-quality estimation of multiple intermediate frames for video interpolation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 9000–9008. [CrossRef]
34. Katz, M.L.; Nikolic, K.; Delbruck, T. Live demonstration: Behavioural emulation of event-based vision sensors. In Proceedings of the 2012 IEEE International Symposium on Circuits and Systems (ISCAS), Seoul, Republic of Korea, 20–23 May 2012; pp. 736–740. [CrossRef]
35. Dong, Y.; Shen, X.; Jiang, Z.; Wang, H. Recognition of imbalanced underwater acoustic datasets with exponentially weighted cross-entropy loss. *Appl. Acoust.* **2021**, *174*, 107740. [CrossRef]
36. Gao, F.; Li, B.; Chen, L.; Shang, Z.; Wei, X.; He, C. A softmax classifier for high-precision classification of ultrasonic similar signals. *Ultrasonics* **2021**, *112*, 106344. [CrossRef] [PubMed]
37. Khan, M.A.; Sharif, M.; Akram, T.; Damaševičius, R.; Maskeliūnas, R. Skin lesion segmentation and multiclass classification using deep learning features and improved moth flame optimization. *Diagnostics* **2021**, *11*, 811. [CrossRef] [PubMed]

38. Meng, Q.; Xiao, M.; Yan, S.; Wang, Y.; Lin, Z.; Luo, Z.Q. Training High-Performance Low-Latency Spiking Neural Networks by Differentiation on Spike Representation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 12444–12453. [CrossRef]
39. Tang, G.; Shah, A.; Michmizos, K.P. Spiking neural network on neuromorphic hardware for energy-efficient unidimensional slam. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 4176–4181. [CrossRef]
40. Imambi, S.; Prakash, K.B.; Kanagachidambaresan, G.R. PyTorch. In *Programming with TensorFlow*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 87–104.
41. Loizou, N.; Vaswani, S.; Laradji, I.H.; Lacoste-Julien, S. Stochastic polyak step-size for sgd: An adaptive learning rate for fast convergence. In Proceedings of the International Conference on Artificial Intelligence and Statistics, Palermo, Italy, 26–28 August 2020; pp. 1306–1314. [CrossRef]
42. Ye, Y.; Tian, Y.; Huenerfauth, M.; Liu, J. Recognizing American sign language gestures from within continuous videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–23 June 2018. [CrossRef]
43. Zhang, Y.; Cao, C.; Cheng, J.; Lu, H. EgoGesture: A new dataset and benchmark for egocentric hand gesture recognition. *IEEE Trans. Multimed.* **2018**, *20*, 1038–1050. [CrossRef]
44. Monti, F.; Boscaini, D.; Masci, J.; Rodola, E.; Svoboda, J.; Bronstein, M.M. Geometric deep learning on graphs and manifolds using mixture model CNNs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5115–5124. [CrossRef]
45. Liu, Z.; Wang, L.; Wu, W.; Qian, C.; Lu, T. TAM: Temporal adaptive module for video recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 13708–13718. [CrossRef]