Event Camera-Based Real-Time Gesture Recognition for Improved Robotic Guidance

AITSAM, Muhammad, DAVIES, Sergio and DI NUOVO, Alessandro <http://orcid.org/0000-0003-2677-2650>

Event Camera-Based Real-Time Gesture Recognition for Improved Robotic Guidance

Muhammad Aitsam
Department of Computing
Sheffield Hallam University
Sheffield, UK
m.aitsam@shu.ac.uk

Sergio Davies
Department of Computing & Mathematics
Manchester Metropolitan University
Manchester, UK
sergio.davies@mmu.ac.uk

Alessandro Di Nuovo
Department of Computing
Sheffield Hallam University
Sheffield, UK
a.dinuovo@shu.ac.uk

***Abstract*—Recent breakthroughs in event-based vision, driven by the capabilities of high-resolution event cameras, have significantly improved human-robot interactions. Event cameras excel in managing dynamic range and motion blur, seamlessly adapting to various environmental conditions. The research presented in this paper leverages this technology to develop an intuitive robot guidance system capable of interpreting hand gestures for precise robot control. We introduce the "EB-HandGesture" dataset, an innovative high-resolution hand-gesture dataset used in conjunction with our network "ConvRNN" to demonstrate commendable accuracy of 95.7% in the interpretation task, covering six gesture types in different lighting scenarios. To validate our framework, real-life experiments were conducted with the ARI robot, confirming the effectiveness of the trained network during the various interaction processes. This research represents a substantial leap forward in ensuring safer, more reliable and more efficient human-robot collaboration in shared workspaces.**

***Index Terms*—event-based, gesture recognition, robot control, gesture dataset**

I. Introduction

Gesture recognition provides a natural and intuitive way for humans to interact with robots, carrying both theoretical importance and practical value. As gesture recognition technology advances, it becomes increasingly useful in the field of Human-Robot Interaction (HRI). However, there is a challenge when it comes to collecting gesture data using regular RGB cameras that have a frame rate of 30-60 frames per second (fps). This method often results in motion blur issues when capturing fast gestures [1], affecting the accuracy of gesture recognition. One common solution to the motion blur problem is to increase the frame rate of the standard camera. However, this approach generates a lot of static and unnecessary information in the continuous image frames. Additionally, it records the background environment, which isn't always needed for gesture recognition. Furthermore, frame-based cameras are ineffective in extreme lighting conditions, such as very bright or dark environments. In HRI, where the robot could be placed in extremely bright or very dark environments, it is important to recognize human gestures for seamless interaction.

This work is funded by Marie Sklodowska-Curie Action Horizon 2020 (Grant agreement No. 955778) for the project 'Personalized Robotics as Service Oriented Applications (PERSEO)'.

Event cameras are the solution to the above-mentioned problems. Designed to mimic biological sensors, event cameras offer several advantages, including removing unnecessary data, quick sensing abilities, high dynamic range sensitivity, and low power consumption. They utilize frameless sampling, allowing for the collection of continuous and asynchronous spatiotemporal data. This demand-driven approach records only the changes in lighting caused by gestures, while eliminating irrelevant background details, resulting in lower transmission bandwidth requirements. Moreover, event cameras offer microsecond time resolution, ensuring the smooth capture of gesture motion without being constrained by exposure time or frame rate [2]. Additionally, they perform well in both well-lit and dimly-lit environments due to their high dynamic range capabilities. Since their introduction, event cameras have found applications in various scenarios within the fields of computer vision and robotics [3] [4]. This paper primarily explores a real-time gesture recognition method based on event camera technology.

In this study, we developed a real-time hand gesture recognition system. To begin, we acquired dynamic gesture data using the CenturyArk SilkyCam VGA event camera. Subsequently, we carried out the essential task of converting the three-dimensional event stream. Following this, we meticulously labeled the dataset and transformed it into the HDF5 format to optimize it for training purposes. Then we trained and fine-tuned our classification model. Our chosen training model was based on the ConvRNN architecture [5] [6]. Finally, we conducted real-time experiments using the event camera and humanoid robot. The event camera demonstrated exceptional performance in different light conditions, making it a valuable tool in situations where standard cameras are less effective. By leveraging the benefits of event cameras, this research showcases the potential of event cameras to improve gesture recognition systems, especially in challenging lighting environments. Figure 1 shows the block diagram of the proposed system.

The remainder of the paper is structured this way. Section II presents an analysis of related work, elucidating the fusion of event cameras with gesture recognition paradigms. Section III briefly discusses the working principle of an event camera. Moving forward, Section IV discusses the methodology

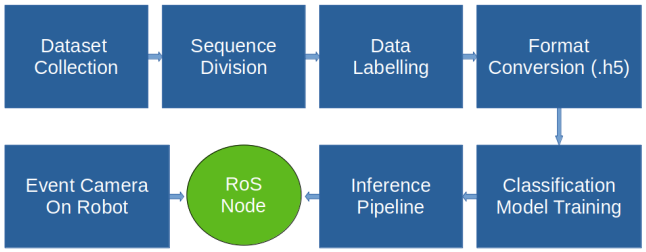

Fig. 1. Block diagram of the proposed system. Starting with data collection to real-time gesture recognition with an event camera mounted on a robot.

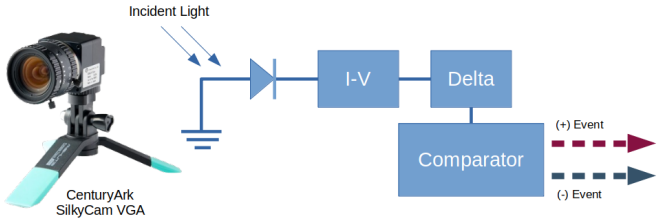

Fig. 2. CenturyArk SilkyCam VGA (event camera) and its basic circuit.

employed in creating a dataset and labeling it. Section V is about how we performed the training and the architecture of the classifier. Looking ahead, in Section VI we presented and discussed the results. Section VII is about the potential applications of the proposed system.

II. Related Work

Gesture recognition systems operating in real-time vary widely in terms of hardware configurations and the algorithms they employ for tasks like gesture classification and localisation. An overview of these algorithms for hand gesture recognition, utilizing both RGB and RGB-D data sources, can be found in a previous study [7].

The use of Convolutional Neural Networks (CNNs) for hand gesture recognition has gained prominence in computer vision applications [8]. CNNs excel at extracting hierarchical features from visual data, making them well-suited for capturing the intricate patterns and variations found in hand gestures [9] [10]. Through deep learning techniques, these models achieve high-accuracy gesture recognition, enabling applications in sign language interpretation, human-computer interaction, and virtual reality [11]. Considerable research efforts have been dedicated to advancing hand gesture recognition techniques [12] [13]. In a notable 2017 work by Amir et al. [14], a pioneering gesture recognition system was developed, which was fully implemented on event-based hardware. This system utilized a TrueNorth neurosynaptic processor to achieve real-time, energy-efficient recognition of hand gestures. This achievement was made possible by processing events streamed from a Dynamic Vision Sensor (DVS), a novel approach that marked a significant milestone in the field. The authors also introduced a proprietary dataset comprising hand gesture samples from a 128x128 DVS camera.

Despite these advancements, there is a significant research gap in the application of gesture recognition systems in robotics. Additionally, the availability of high-resolution hand gesture datasets is limited, which hinders comprehensive investigations. It's important to note that access to neuromorphic processors (e.g., SpiNNaker [15], Loihi [16], TrueNorth [17]) remains constrained due to their limited commercial availability. Consequently, there is a growing need for a system capable of employing commercially available event cameras for gesture recognition, without the necessity for neuromorphic hardware. Lastly, so far there is no system available that uses an event camera for robot control.

In this article, we introduce a new high-resolution hand gesture dataset (640x480). Leveraging this dataset, we trained a ConvRNN model, achieving commendable accuracy rates. By integrating an event camera onto a robotic platform, we enable real-time hand gesture recognition, thus enabling precise robot control based on recognized gestures. This endeavor addresses crucial gaps in both high-resolution gesture datasets and the application of event camera-based gesture recognition systems in robotics.

III. Event Camera

In this study, we utilized the SilkyCam VGA event camera, manufactured by CenturyArk, as our primary imaging device [18]. The SilkyCam VGA event camera is equipped with Prophesee's Metavision software stack to carry out its operations. Figure 2 shows the event camera and its basic circuit diagram. When light hits each pixel, it is transformed into a voltage. This voltage change from the reference level is identified, and if this change surpasses a certain threshold in a comparator, an event is then generated.

The standard notation used to represent an event is as follows:

$$e = [x, y, t, p]$$

Here, event e signifies that the pixel situated at coordinates [x, y] within the pixel array of the event camera emitted an event in response to an illumination change at time t. The polarity attribute is encoded as p=[0, 1], with p=1 denoting an ON event and p=0 representing an OFF event. It is noteworthy that these events are transmitted at a temporal resolution of 1 µs [19], and the data rate of events depends on the rate of illumination changes occurring in the scene. Most of the companies use their proprietary data formats to save the event data. For example, Prophesee saves data in EVT (2.0, 3.0) format and iniVation saves in AEDAT (3.0, 3.1, 4.0) format.

IV. Event Dataset

As of now, most of the event-based gesture datasets are either converted/reproduced from the standard camera dataset or they are created with low-resolution datasets. Datasets such as N-MNIST, N-Caltech101 [22], CIFAR10-DVS [20], MNIST-DVS [21] and N-ImageNet [24] are recorded by moving an event camera around monitors displaying images

TABLE I
COMPARISON OF DATASETS

Dataset Name	Year	Type	Data	Sensor	Resolution	Sec. per Instance	Samples	Reference
CIFAR10-DVS	2017	reproduced	images	DAVIS128	128x128	1.2s	10k	[20]
MNIST-DVS	2013	reproduced	digit images	DAVIS128	128x128	2-3s	30k	[21]
N-MNIST	2015	reproduced	digit images	ATIS	28x28	0.3s	70k	[22]
N-CALTECH101	2015	reproduced	images	ATIS	302x245	0.3s	8.7k	[22]
ES-ImageNet	2021	converted	images	-	224x224	-	1.3mil	[23]
N-ImageNet	2021	reproduced	images	Samsung Gen3	480x640	-	1.7mil	[24]
HARDVS	2022	event-based	action	DAVIS346	346x260	5s	100k	[25]
Daily Action	2021	event-based	action	DAVIS346	346x260	5s	1.4k	[26]
Bullying 10K	2023	event-based	action	DAVIS346	346x260	2-20s	10k	[27]
ASL-DVS	2019	event-based	hand action	DAVIS240	240x180	0.1s	100k	[28]
Nav-DVS	2020	event-based	hand action	ATIS	302x245	-	1.3k	[29]
DVS128 Gesture	2017	event-based	hand action	DAVIS128	128x128	6s	1.3k	[14]
EB-HandGesture	2024	event-based	hand action	SilkyCam Gen3	640x480	0.5s	1.5k	This work

of well-known datasets like MNIST [30], Caltech101 [31] and ImageNet [32]. Yihan lin et al. [23] used an Omnidirectional Discrete Gradient (ODG) algorithm to convert the ImageNet dataset into its event-stream (ES) version. They called it ES-ImageNet.

A variety of event-based datasets are publicly accessible, as highlighted in recent surveys [33]. This article focusses primarily on datasets geared towards recognition tasks. Datasets such as HARDVS [25], Daily Action [26], and Bullying10k [27] pertain to human actions, while others like IBM-DVS128 Gesture [14], ASL-DVS [28], and Nav-DVS [29] are centered on hand gestures. Commonly used event cameras for creating these datasets include DAVIS128 (128x128), DAVIS240 (240x180), DAVIS346 (346x260) [34], and ATIS (302x245) [35]. The resolution of these event cameras is a critical factor affecting their performance. Higher-resolution cameras offer significant advantages, such as capturing finer details within a scene. This enhanced spatial resolution is crucial for tasks requiring precise object tracking, detailed motion analysis, or intricate scene reconstruction. Furthermore, even with brief accumulation times, high-resolution event cameras can effectively capture sufficient information for gesture recognition. Second, with advances in this area, new high-resolution event cameras are entering the market [19]. This development poses a challenge for researchers, as they cannot use old, low-resolution datasets with new event cameras. For instance, one of the newer sensors, CenturyArks' SilkyCam VGA, has a resolution of 640x480 and can only accept inputs divisible by this resolution (e.g., 320x240, 160x120). However, one widely used event-based hand gesture dataset, the IBM-DVS128Gesture (captured with iniVation DVS128 event camera), has a resolution of 128x128. Consequently, even if a user trains an effective classification model for this dataset, it cannot perform real-time inference with SilkyCam VGA due to the mismatch in input resolution.

We introduce the first high-resolution hand gesture dataset, named the EB-HandGesture dataset, created using the CenturyArk SilkyCam Gen3.0 (640x480). This camera is outfitted with a Prophesee event-based vision sensor, boasting a temporal resolution of 1μsec. To collect data, we utilized the

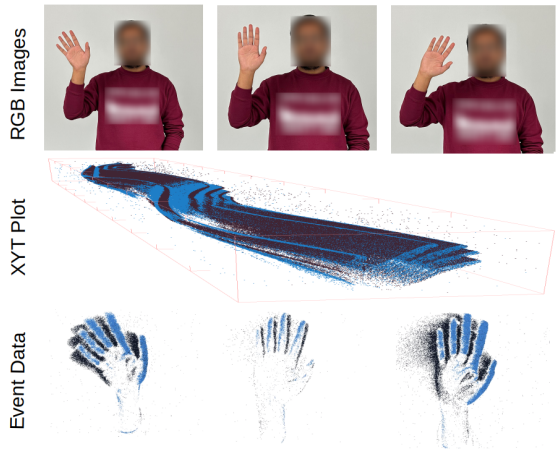

Fig. 3. The system discussed in this paper processes data displayed in the final row, illustrating frame-based and event-based camera outputs. The top section shows RGB images of a hand gesture (wave), the middle section depicts positive (blue) and negative (black) DVS events over time, and the bottom section presents the DVS event data corresponding to the executed gesture.

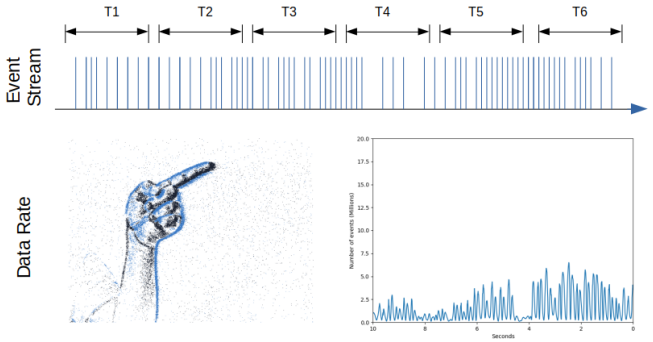

Fig. 4. (top) event stream sequencing. (bottom) illustration of performed gestures and the number of events captured during time.

Prophesee Metavision SDK and the OpenEB [6] framework. Our dataset includes ground-truth files featuring gesture labels along with their start and stop times, all gathered through labeling system designed specifically for event data. The EB-HandGesture dataset encompasses 9000 instances across 6 hand gestures, contributed by 5 participants. These gestures, recorded at three different speeds (slow, normal, fast) and under two lighting conditions (normal, low), include hand waves, pointing, rock, scissors, claps, and arm rolls. Each instance has a duration of 0.5 seconds, amounting to 1500 instances for each gesture. The dataset's details are summarized in Table I, alongside other event-based recognition datasets. Figure 3 shows the comparison between RGB images and event data along with the XYT plot of the performed gesture. Figure 4 demonstrates the sequencing of the event stream (top) and data rate for the gesture performed (bottom).

TABLE II
CONVRNN CLASSIFIER MODEL ARCHITECTURE

Layer Type	In Ch.	Out Ch.	Stride	Remarks
ConvLayer	1	16	1	Initial conv
ConvLayer	16	32	2	Down-sample
ConvLayer	32	32	1	Same depth
ConvLayer	32	64	2	Increase ch.
ConvLayer	64	64	1	Pre-recurrent
ConvRNN	64	128	2	First ConvRNN
ConvRNN	128	256	2	Second ConvRNN
Conv2d	256	256	1	Pre-output
Conv2d	256	6	1	Output

V. MODEL TRAINING

In this section, we describe the training process for our Convolutional Recurrent Neural Network (ConvRNN) Classifier, which serves as the backbone of our hand gesture recognition system [36]. We provide insights into the dataset preparation, model architecture, and training methodology.

A. Dataset Preparation

We utilized the EB-HandGesture dataset, a novel collection of event-based data specifically designed for hand gesture recognition and neuromorphic classification. The dataset was partitioned into training (70%), validation (20%), and testing (10%) subsets. With its unique characteristics, we believe that the EB-HandGesture dataset offers an exceptional platform for evaluating neuromorphic classification models, particularly for hand gesture recognition tasks in the context of robotic control.

To facilitate data preprocessing and training, we harnessed the capabilities of the Prophesee OpenEB open-source project, tightly integrated with the Metavision SDK. The SilkyCam Gen3.0 event camera was employed to capture data in EVT3.0 format. Additional information about this data format can be found in the Prophesee documentation [6]. This data is then converted to avi video format for annotation and also converted to HDF5 format to make it ready for training.

B. Model Architecture and Training

The ConvRNN classifier that we are proposing is designed to accommodate sequential data and has demonstrated remarkable performance in our experiments. The architecture of this classifier is detailed in Table II.

The ConvRNN Classifier comprises several integral components. The input data, consisting of a single channel, undergoes initial processing through a series of convolutional layers. The number of these layers is determined by the size of the input, allowing for the extraction of relevant features. Following feature extraction, the data is subjected to two levels of ConvRNN layers. These layers incorporate recurrent connections, enabling the model to capture temporal dependencies within the data. The final segment of the ConvRNN Classifier is the classification head. It consists of additional convolutional layers, followed by Rectified Linear Unit (ReLU) activation functions. This section ultimately produces class probabilities, serving as the model's output. Each event stream corresponds to a single HDF5 file and an accompanying .npy file containing ground truth data. During training, we employed a batch size of 32, initiated the learning rate at 0.0001, and utilized the Adam optimizer. Our implementation was executed in Python 3.8, leveraging the computational power of an RTX2080 GPU.

C. Evaluation Matrix

In addition to training our ConvRNN Classifier, we conducted comparative experiments by training our dataset with alternative classifiers, including MobileNetV2 [37] and SqueezeNet [38]. Moreover, we also the used SpikeBased-BP algorithm to train a classifier for our dataset. These comparisons allow us to evaluate the effectiveness of our proposed model against existing architectures.

The network's performance is assessed by determining the accuracy of its outputs to the given labels. However, using accuracy as the sole metric for evaluation might not sufficiently capture the model's proficiency in accurately predicting the final class. To remedy this, we have implemented the use of the Precision-Recall (PR) curve [39]. This curve facilitates a more nuanced examination of the model's predictive abilities under different conditions.

VI. RESULTS AND DISCUSSION

Figure 5 (a) Left: visually represents the training and validation accuracy (y-axis) for each epoch (x-axis). As training progresses, we observe a consistent upward trajectory in both accuracies, with getting commendable accuracy of 96.9% (training) and 96.2% (validation). Right: shows the accuracy of each class to epoch. At epoch=50 the accuracy of each class is more than 93%. (b) shows the confusion and error matrices for a better understanding of the results.

Figure 5 (c) shows Precision-Recall curves for all the classes. The P-R curves for the classes 'wave', 'clap', 'arm-roll', 'point', 'rock', and 'scissor' demonstrate varying levels of classification performance by the model. Classes like 'wave', 'clap', and 'scissor' show high precision and recall, with only a slight decline in precision at higher recall levels,

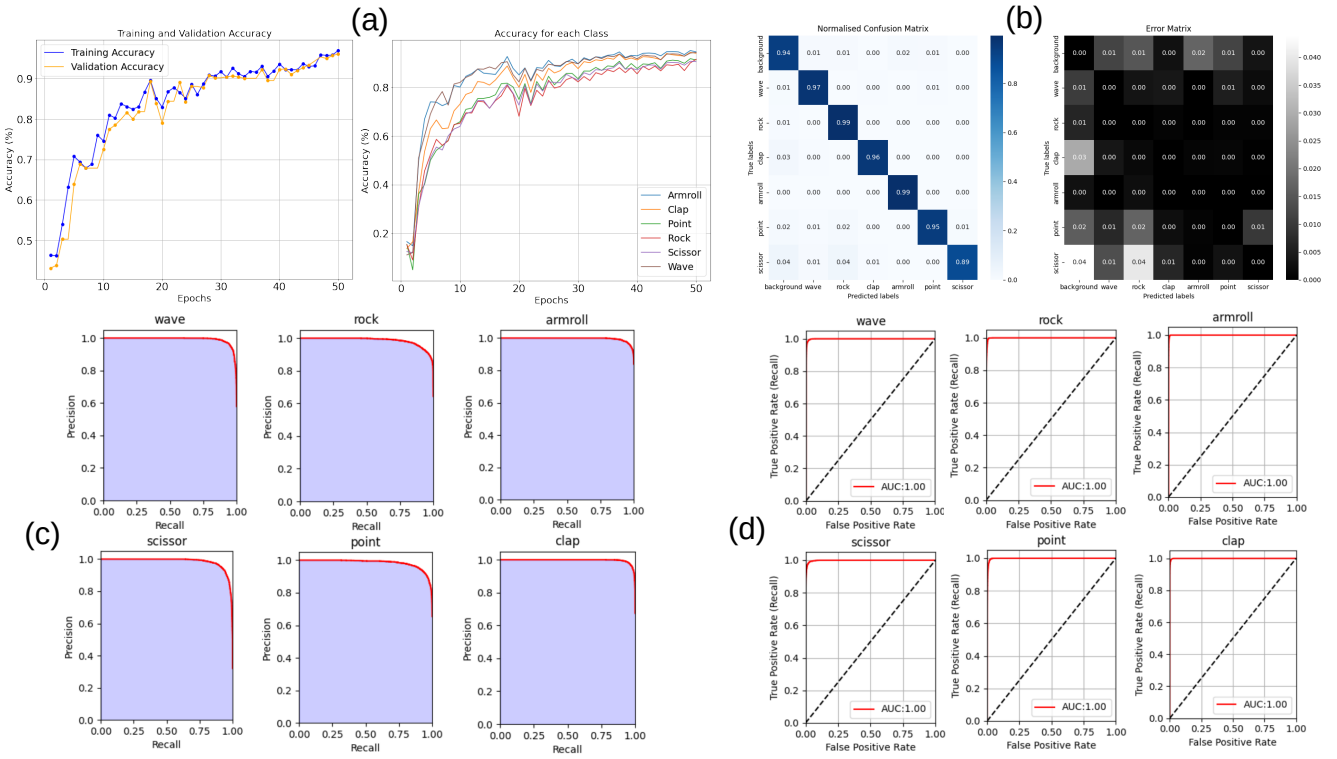

Fig. 5. (a) left: training and validation accuracy (y-axis) for each epoch (x-axis). right: Validation accuracy of each class. (b) Confusion and Error Matrix. (c) Precision (y-axis), Recall (x-axis) curve. (d) Receiver Operating Characteristic (ROC) Curves for Gesture Recognition Model.

indicating the model's effectiveness in identifying these gestures with minimal false positives. The 'rock' class exhibits more fluctuation, with precision decreasing as recall increases, suggesting a higher rate of false positives when the model attempts to identify all true positives. 'Armroll' and 'point' maintain high precision across most recall levels but display a minor dip at higher recalls, pointing to an increase in false positives as the model strives for higher recall. The 'clap' class appears to have a perfect precision across all recall levels, which could indicate either outstanding model performance or a potential issue with data completeness or graph rendering. (d) shows the ROC curves plot the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings. The Area Under the Curve (AUC) for all models is 1.00, indicating perfect discrimination by the models with no overlap between the positive and negative distributions.

In Table III we compared the existing event-based datasets related to action and gesture with our dataset. For each dataset, we showed the results of state-of-art models. In general, we got high testing accuracy for our dataset as compared to other datasets and their models. As DVS128Gesture is closest to the EB-HandGesture dataset, we trained the DVS128Gesture dataset with our proposed model. Additionally, for comparison purposes, we used the SpikeBased-BP algorithm to train EB-HandGesture. For the ConvRNN Classifier, the maximum accuracy we got for the DVS128Gesture dataset was 87.42%, which is much lower than the EB-HandGesture dataset (95.58%). For SpikeBased-BP, the accuracy for DVS128Gesture was 95.5% and for EB-HandGesture it was 85.6%. We also compared our model with MobileNet (70.44%) and Sqeezenet (75.65%). We also created a custom RGB Gesture Dataset and trained it with LSTM to compare the results. Although the accuracy of LSTM is high it does not perform well because of RGB camera limitations. Figure

TABLE III
COMPARISON OF VARIOUS MODELS AND DATASETS

Ref.	Dataset	Algo./Model	Accuracy(%)
[40]	ASL-DVS	G-CNN	87.5
[40]		RG-CNN	90.1
[41]		MobileNet	86.7
[42]	HAR-DVS	ESTF	57.53
[43]		ResNet18	56.09
[44]		ResNet50	57.99
[45]	Daily Action	Motion SNN	90.3
[46]		HMAX-SNN	76.9
[47]	Bullying10k	ResNet50	74.01
[48]		ResNet18	72.5
[49]		X3D	65.6
[50]	DVS128Gesture	Deep-SNN	93.6
[51]		ConvRNN-SNN	90.28
[52]		SpikeBases-BP	95.5
This Work		ConvRNN	87.42
This Work	EB-HandGesture	MobileNet	70.44
This Work		SqueezeNet	75.65
This Work		ConvRNN	**95.77**
This Work		SpikeBased-BP	85.6
This Work	Custom RGB Dataset	LSTM	98.9

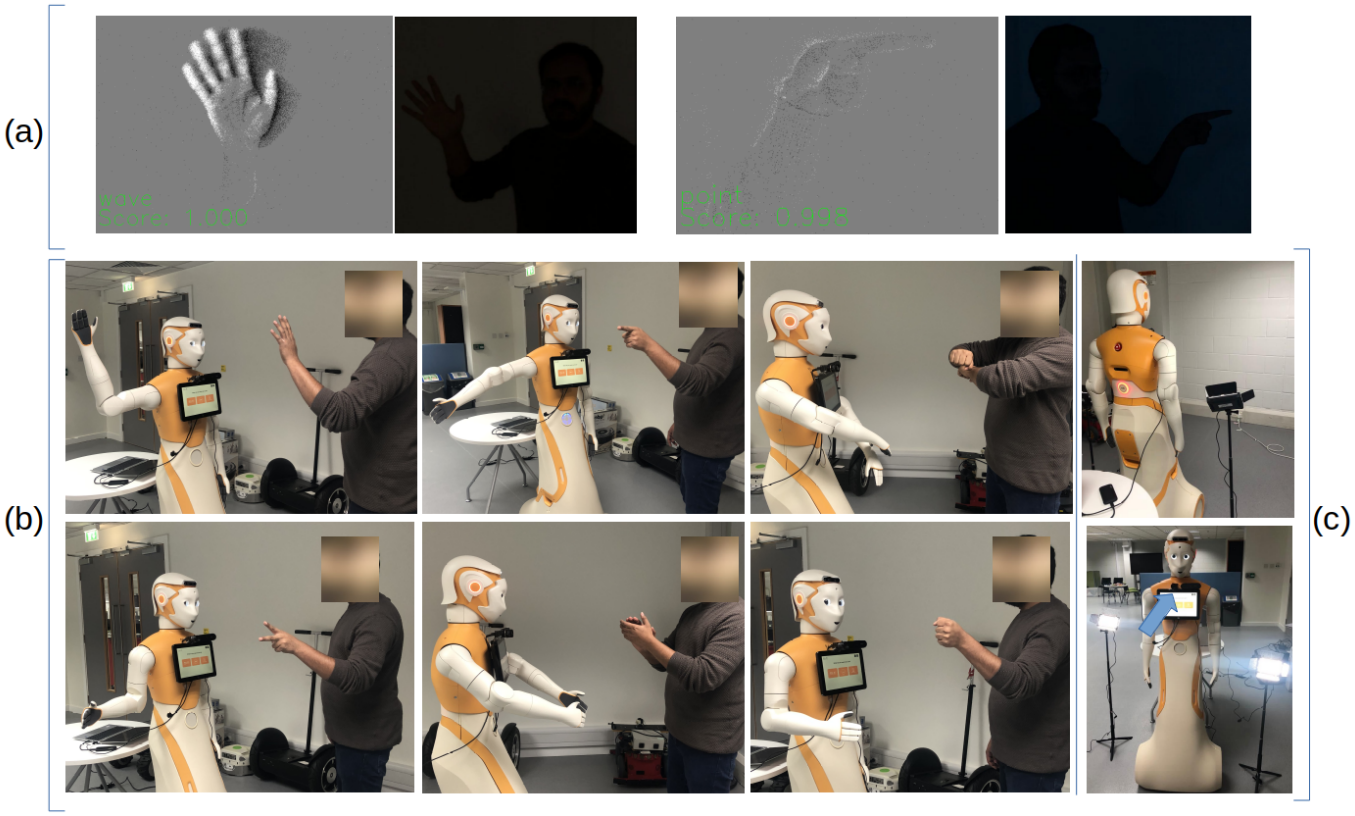

Fig. 6. (a) Testing our model in low light conditions where the standard camera was not able to detect. (b) real-time experiments with an event camera and ARI robot for different hand gestures. (c) data collection setup. The arrow is pointing toward the mounted event camera.

6(a) demonstrates the superior low-light performance of the event camera in scenarios where standard cameras fail to detect any activity. The robustness of our model in these conditions enhances the reliability of human-robot interactions, as it surpasses current state-of-the-art models in accurately interpreting gestures in challenging conditions.

A. Robot Control

We demonstrated the real-world applicability of our gesture recognition model through the use of the ARI humanoid robot and an event camera for intuitive robot control. The event camera was mounted on the ARI robot, enabling seamless integration of the classifier with the robot's control system. This integration permitted us to manipulate the robot's movements using predicted hand gestures. Our experimental procedure entailed initializing the robot, activating the event camera, and starting the classification pipeline. For real-time classification, we segmented the input event stream into 1-second intervals. We associated each gesture with a custom robot movement. The model and inference pipeline can be used to associate gestures with any movement according to the application. Figure 6 (b) the ARI robot recognizing a hand gesture. Here for *wave* gesture, ARI is waving. For *point*, it is showing direction. For *rock*, it is showing paper. For *clap* and *arm roll* it is trying to imitate the gestures. (c) shows the data collection setup.

Visit the project website for more details and results comparison: *https://sites.google.com/view/event-camera-based-gesture/home*

VII. Potential Applications

The potential applications of our hand gesture recognition system with an event camera in robot control are particularly promising in sectors such as healthcare and collaborative work environments. In healthcare, especially in elder care, the system allows elderly individuals to command robots using simple hand gestures. This feature is invaluable for those with limited mobility or communication abilities, as the event camera accurately captures gestures without the issues of motion blur that standard cameras face. This ensures reliable and precise operation, crucial in healthcare settings where speed and accuracy are essential.

In collaborative workspaces such as warehouses, this technology can significantly improve the interaction between humans and robots. Workers can use hand gestures to direct robots, a method that is especially beneficial in environments where verbal communication might be impractical or ineffective. The robustness of the system in various lighting conditions guarantees consistent performance, thereby enhancing productivity and fostering a more reliable human-robot working relationship. Overall, this event-based gesture-controlled robotic system could greatly enhance operational efficiency

and user experience across multiple sectors, representing a significant step forward in this area.

VIII. Conclusion

Our study marks a significant advancement in event-based vision and human-robot interaction, harnessing the capabilities of high-resolution event cameras for hand gesture recognition. The consistent improvements in accuracy, reaching an impressive testing accuracy of 95.77%, underscore the exceptional performance of our ConvRNN model. We put our model to the test in the real world by seamlessly integrating it with an ARI robot. This integration enables effortless and precise control of the robot through hand gestures. The successful deployment of our model in various environmental conditions highlights its adaptability and its potential to enhance human-robot teamwork.

Acknowledgment

This work is funded by Marie Sklodowska-Curie Action Horizon 2020 (Grant agreement No. 955778) for the project 'Personalized Robotics as Service Oriented Applications (PERSEO)'. The authors would like to thank lab members for their help and support. For open access, the author has applied a Creative Commons Attribution (CC BY) license to any Author Accepted Manuscript version arising from this submission.

References

[1] M. Tico, N. Gelfand, and K. Pulli, "Motion-blur-free exposure fusion," *Proceedings - International Conference on Image Processing, ICIP*, pp. 3321–3324, 2010.

[2] T. Stoffregen and L. Kleeman, "Event cameras, contrast maximization and reward functions: an analysis," 2019.

[3] K. Golibrzuch, S. Schwabe, T. Zhong, K. Papendorf, and A. M. Wodtke, "Application of an event-based camera for real-time velocity resolved kinetics," *Journal of Physical Chemistry A*, vol. 126, pp. 2142–2148, 4 2022.

[4] S. Jia, "Event camera survey and extension application to semantic segmentation," *ACM International Conference Proceeding Series*, pp. 115–121, 3 2022. [Online]. Available: https://dl.acm.org/doi/10.1145/3529446.3529465

[5] F. M. Bianchi, E. Maiorino, M. C. Kampffmeyer, A. Rizzi, and R. Jenssen, *Recurrent neural network architectures*. Springer, 2017, vol. 0, pp. 23–29.

[6] Prophesee, "Evt 3.0 format — metavision sdk docs 4.3.0 documentation," 2023. [Online]. Available: $https://docs.prophesee.ai/stable/data/encoding_formats/evt3.html$

[7] P. K. Pisharady and M. Saerbeck, "Recent methods and databases in vision-based hand gesture recognition: A review," *Computer Vision and Image Understanding*, vol. 141, pp. 152–165, 12 2015.

[8] D. Bhatt, C. Patel, H. Talsania, J. Patel, R. Vaghela, S. Pandya, K. Modi, and H. Ghayvat, "Cnn variants for computer vision: History, architecture, application, challenges and future scope," 10 2021.

[9] O. Koller, H. Ney, and R. Bowden, "Deep hand: How to train a cnn on 1 million hand images when your data is continuous and weakly labelled," 2016. [Online]. Available: http://www.hltpr.rwth-aachen.de/

[10] I. C. Society, I. of Electrical, and E. Engineers, *Computer Vision and Pattern Recognition Workshops (CVPRW), 2015 IEEE Conference on : date, 7-12 June 2015.*, 2015.

[11] N. Neverova, C. Wolf, G. W. Taylor, and F. Nebout, "Moddrop: adaptive multi-modal gesture recognition," 12 2014. [Online]. Available: http://arxiv.org/abs/1501.00102

[12] P. G. Veronica, R. K. Mokkapati, L. P. Jagupilla, and C. Santhosh, "Static hand gesture recognition using novel convolutional neural network and support vector machine," *International journal of online and biomedical engineering*, vol. 19, pp. 131–141, 2023.

[13] A. Alnuaim, M. Zakariah, W. A. Hatamleh, H. Tarazi, V. Tripathi, and E. T. Amoatey, "Human-computer interaction with hand gesture recognition using resnet and mobilenet," *Computational Intelligence and Neuroscience*, vol. 2022, 2022.

[14] A. Amir, B. Taba, D. Berg, T. Melano, J. Mckinstry, C. D. 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," vol. 2017-January. Institute of Electrical and Electronics Engineers Inc., 11 2017, pp. 7388–7397.

[15] C. Mayr, S. Hoeppner, and S. Furber, "Spinnaker 2: A 10 million core processor system for brain simulation and machine learning," 11 2019. [Online]. Available: http://arxiv.org/abs/1911.02385

[16] R. B. Uludağ, S. Çağdaş, Y. S. İşler, N. S. Şengör, and I. Akturk, "Bio-realistic neural network implementation on loihi 2 with izhikevich neurons," 7 2023. [Online]. Available: http://arxiv.org/abs/2307.11844

[17] F. Akopyan, J. Sawada, A. Cassidy, R. Alvarez-Icaza, J. Arthur, P. Merolla, N. Imam, Y. Nakamura, P. Datta, G. J. Nam, B. Taba, M. Beakes, B. Brezzo, J. B. Kuang, R. Manohar, W. P. Risk, B. Jackson, and D. S. Modha, "Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, pp. 1537–1557, 10 2015.

[18] CenturyArk, "Silkyevcam (vga) - centuryarks co., ltd." [Online]. Available: https://centuryarks.com/en/silkyevcam-vga/

[19] E. Perot, P. de Tournemire, D. Nitti, J. Masci, and A. Sironi, "Learning to detect objects with a 1 megapixel event camera," 9 2020. [Online]. Available: http://arxiv.org/abs/2009.13436

[20] H. Li, H. Liu, X. Ji, G. Li, and L. Shi, "Cifar10-dvs: An event-stream dataset for object classification," *Frontiers in Neuroscience*, vol. 11, 5 2017.

[21] T. Serrano-Gotarredona and B. Linares-Barranco, "A 128,× 128 1.5

[22] G. Orchard, A. Jayawant, G. K. Cohen, and N. Thakor, "Converting static image datasets to spiking neuromorphic datasets using saccades," *Frontiers in Neuroscience*, vol. 9, 2015.

[23] Y. Lin, W. Ding, S. Qiang, L. Deng, and G. Li, "Es-imagenet: A million event-stream classification dataset for spiking neural networks," 10 2021. [Online]. Available: http://arxiv.org/abs/2110.12211 http://dx.doi.org/10.3389/fnins.2021.726582

[24] J. Kim, J. Bae, G. Park, D. Zhang, and Y. M. Kim, "N-imagenet: Towards robust, fine-grained object recognition with event cameras," 2021.

[25] X. Wang, Z. Wu, B. Jiang, Z. Bao, L. Zhu, G. Li, Y. Wang, and Y. Tian, "Hardvs: Revisiting human activity recognition with dynamic vision sensors," 11 2022. [Online]. Available: http://arxiv.org/abs/2211.09648

[26] Q. Liu, D. Xing, H. Tang, D. Ma, and G. Pan, "Event-based action recognition using motion information and spiking neural networks," 2021.

[27] Y. Dong, Y. Li, D. Zhao, G. Shen, and Y. Zeng, "Bullying10k: A neuromorphic dataset towards privacy-preserving bullying recognition," 6 2023. [Online]. Available: http://arxiv.org/abs/2306.11546

[28] Y. Bi, A. Chadha, A. Abbas, E. Bourtsoulatze, and Y. Andreopoulos, "Graph-based object classification for neuromorphic vision sensing," 2019.

[29] J. M. Maro, S. H. Ieng, and R. Benosman, "Event-based gesture recognition with dynamic background suppression using smartphone computational capabilities," *Frontiers in Neuroscience*, vol. 14, 4 2020.

[30] M. Fatahi, "Mnist handwritten digits description and using neuromorphic hardware view project," 2014. [Online]. Available: https://www.researchgate.net/publication/273124795

[31] N. Kamarudin, M. Makhtar, F. S. Abdullah, M. Mohamad, N. S. Kamarudin, S. A. Fadzli, F. S. Mohamad, and M. F. A. Kadir, "Comparison of image classification techniques using caltech 101 dataset texture-based image retrieval view project mobile quranic memorization tool view project comparison of image classification techniques using caltech 101 dataset," *Article in Journal of Theoretical and Applied Information Technology*, vol. 10, 2015. [Online]. Available: www.jatit.org

[32] J. Deng, W. Dong, R. Socher, L.-J. Li, and L. Fei-Fei, *Computer Vision and Pattern Recognition, 2009, CVPR 2009, IEEE Conference on : dates: 20-25 June 2009.* IEEE, 2009.

[33] G. Gallego, T. Delbruck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, "Event-based vision: A

survey," 4 2019. [Online]. Available: http://arxiv.org/abs/1904.08405 http://dx.doi.org/10.1109/TPAMI.2020.3008413

[34] E. Mueggler, H. Rebecq, G. Gallego, T. Delbruck, and D. Scaramuzza, "The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and slam," *International Journal of Robotics Research*, vol. 36, pp. 142–149, 2 2017.

[35] X. Clady, J.-M. Maro, S. E. B. barre, and R. Benosman, "Frontiers in neuromorphic engineering a motion-based feature for event-based pattern recognition."

[36] OpenEB and Prophesee, "Training an eb classification model — metavision sdk docs 4.5.1 documentation." [Online]. Available: $https://docs.prophesee.ai/stable/tutorials/ml/training/train_classifier.html$

[37] K. Dong, C. Zhou, Y. Ruan, and Y. Li, "Mobilenetv2 model for image classification." Institute of Electrical and Electronics Engineers Inc., 12 2020, pp. 476–480.

[38] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and ¡0.5mb model size," 2 2016. [Online]. Available: https://arxiv.org/abs/1602.07360v4

[39] C. K. I. Williams, "The effect of class imbalance on precision-recall curves," 2020.

[40] Y. Bi, A. Chadha, A. Abbas, E. Bourtsoulatze, and Y. Andreopoulos, "Graph-based object classification for neuromorphic vision sensing," 2019.

[41] F. Monti, D. Boscaini, J. Masci, E. Rodolà, J. Svoboda, and M. M. Bronstein, "Geometric deep learning on graphs and manifolds using mixture model cnns," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 5425–5434, 11 2017.

[42] X. Wang, Z. Wu, B. Jiang, Z. Bao, L. Zhu, G. Li, Y. Wang, and Y. Tian, "Hardvs: Revisiting human activity recognition with dynamic vision sensors," 2022. [Online]. Available: https://www.prophesee.ai

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December, pp. 770–778, 12 2016.

[44] Z. Liu, L. Wang, W. Wu, C. Qian, and T. Lu, "Tam: Temporal adaptive module for video recognition," 2023.

[45] Q. Liu, D. Xing, H. Tang, D. Ma, and G. Pan, "Event-based action recognition using motion information and spiking neural networks," 2021.

[46] Q. Liu, H. Ruan, D. Xing, H. Tang, and G. Pan, "Effective aer object classification using segmented probability-maximization learning in spiking neural networks," *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, pp. 1308–1315, 2020.

[47] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," pp. 6202–6211, 2019. [Online]. Available: https://github.com/

[48] D. Tran, J. Ray, Z. Shou, S.-F. Chang, and M. Paluri, "Convnet architecture search for spatiotemporal feature learning," 8 2017. [Online]. Available: https://arxiv.org/abs/1708.05038v1

[49] C. Feichtenhofer, "X3d: Expanding architectures for efficient video recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 200–210, 4 2020. [Online]. Available: https://arxiv.org/abs/2004.04730v1

[50] S. B. Shrestha and G. Orchard, "Slayer: Spike layer error reassignment in time," *Advances in Neural Information Processing Systems*, vol. 31, 2018. [Online]. Available: https://bitbucket.org/bamsumit/slayer

[51] Y. Xing, G. D. Caterina, and J. Soraghan, "A new spiking convolutional recurrent neural network (scrnn) with applications to event-based hand gesture recognition," *Frontiers in Neuroscience*, vol. 14, p. 590164, 11 2020.

[52] J. Kaiser, H. Mostafa, and E. Neftci, "Synaptic plasticity dynamics for deep continuous local learning (decolle)," *Frontiers in Neuroscience*, vol. 14, 11 2018. [Online]. Available: http://arxiv.org/abs/1811.10766 http://dx.doi.org/10.3389/fnins.2020.00424