Frugal event data: how small is too small? A human performance assessment with shrinking data

Amélie Gruel Lucía Trillo Carreras Marina Bueno García Ewa Kupczyk Jean Martinet
i3S / CNRS, Université Côte d'Azur, Sophia Antipolis, France
Contact author: amelie.gruel@univ-cotedazur.fr

Abstract

When designing embedded computer vision systems with limited computational budget, one often needs to take care of the size of input data. In recent years, however, event cameras have shown increasingly large sensor sizes. How small can event data be, while preserving sufficient information for the task at hand? We present in this paper a study to assess and compare human performance in a gesture classification task using event data. Original event data from IBM's DVS128 Gesture dataset is downscaled with several spatial and temporal methods, and the classification performance on 4 classes is measured with human participants. The contributions of this paper are 3-fold: (1) we establish a size threshold under which the human performance falls behind the chance level, (2) we compare several spatial and temporal event downscaling methods and show that all methods give unequal data quality, and (3) we highlight some unexpected discrepancies in a comparison between human vs machine performance. To the best of our knowledge, this is the first human perception study with event data.

1. Introduction

In recent years, event cameras have gained significant attention as an alternative or a complement to traditional frame-based cameras for computer vision applications. Event cameras are capable of capturing temporal changes in the scene with high temporal resolution and low latency, making them ideal for real-time applications [4]. However, event data typically contains much more temporal information than traditional frame-based cameras [10], making it challenging to process and analyse event data on embedded systems with limited computational budgets.

One approach to address this challenge is to downscale the event data before processing it. Downscaling the data can significantly reduce the computational cost while maintaining sufficient information for the intended task [6, 7]. However, the optimal size for downscaling event data is not well established, and there is a need to evaluate the trade-off between data size and task performance. In this context, we present a study to assess and compare human performance in a gesture classification task using event data. The study uses original event data from IBM's DVS128 Gesture Dataset [1], which consists of a collection of hand gestures captured using a dynamic vision sensor (DVS) camera.

The study evaluates human classification performance of four different hand gestures using event data downscaled with several methods. The study measures human performance to establish a size threshold under which human performance falls below the chance level. The paper also compares several downscaling methods and shows that all methods yield unequal data quality. Additionally, unexpected discrepancies in a comparison between human and machine learning are highlighted.

This study is the first of its kind to investigate human perception with event data; this study adds to the machine performance analysis presented in [6] as humans are less susceptible to spurious correlation in the data than most neural networks [13] and do not suffer from shortcut learning [5], especially on the relatively small dataset that is DVS128 gesture (only 133 samples). The findings provide insights into optimising event data for gesture classification tasks, which are crucial in embedded computer vision systems with limited computational budgets, where low latency and energy efficiency are essential.

2. Material and Method

Hand gesture recognition is a skill used daily in human society and is tightly integrated with verbal communication, hence the need for computational learning of such data. Since it relies heavily on low latency, this task is well-suited for event-based computation.

2.1. Event dataset

Building on this notion, Amir et al. presented at CVPR 2017 a complete hand gesture neuromorphic dataset called

DVS128 Gesture [1]. As this dataset has now become a standard benchmark in event data classification, we will assess in this work the evolution of the human performance of gesture classification on downscaled samples. To this end, 29 subjects were recorded performing 11 different hand gestures under 3 kinds of illumination conditions, by a DVS128 camera. A total of about 133 samples are available for each gesture, each composed roughly of 400,000 events and of dimension 128×128 pixels, for a duration of 6 seconds approximately. The dataset is split into two sub-datasets to facilitate machine learning training: the train sub-category received 80% of the recorded samples and the test 20%, with an even distribution of the 11 gestures between them.

In this work, we aim to evaluate the evolution of human performance according to different event downscaling methods at different resolutions — not the human performance in a classification task. To achieve this objective, one did not have to correctly classify each of the 11 gestures comprised in DVS128 Gesture, but would have to demonstrate that they were able to correctly separate very similar gestures in order to prove that the reduction method studied retained sufficient relevant information. For this purpose, we decided to display to the participants only four gestures instead of eleven, two by two similar together but different between the two pairs; our choice was to focus on the gestures "right hand clockwise", "right hand counterclockwise" (both quite similar, only differentiable by the direction of the hand movement), "drums" and "hand rolls" (similar through the spatial use of events). For each figure, a short sample from 5 of the original 29 recorded subjects were selected and randomly displayed to the participant, so that they classify the gesture itself and not the performing subject. Fig. 1 presents a screenshot of those four gestures as presented to the participants.

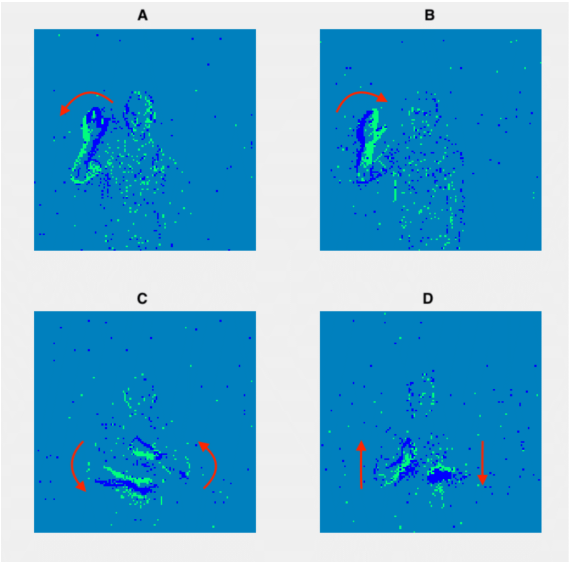

Figure 1. Poster presented to the participants, illustrating the four annotated gestures selected from the DVS128 Gesture dataset to conduct the experiment.

2.2. Methods for event data downscaling

2.2.1 Spatial event downscaling

We describe below the six spatial event downscaling techniques that we assess in this paper. The first four methods were introduced by Gruel et al. in [7]: a simple event funnelling, a method based on a count of positive and negative events and two log-luminance reconstructions, one linear and the other cubic. The additional two neuromorphic methods were introduced by the same authors in [6]. All these methods reduce the data by downscaling the x, y coordinates of pixels, bringing an original $width \times height$ sensor size to a target $(width/ratio) \times (height/ratio)$ size, where $ratio$ is the downscaling ratio.

Spatial funnelling The spatial funnelling method simply consists in dividing all the spatial coordinates of the events by the dividing factor (and removing any duplicate) to obtain the spatially reduced events. From a computational point of view, this downscaling method consists simply in updating the memory address of the event's x, y coordinates. This process can easily be implemented with low resource usage given its simplicity. Other advantages lie in speed and the absence of significant resource usage. However, a main drawback is the increased spatial density in the event data, as nearly every event is kept, which may have an impact on the target task.

Event count The event count method consists in estimating the normalised value reached by the log-luminance related to the (larger) pixels in each target size. This normalised *event count* is updated every time a new event is triggered. By definition, this method waits for the next event to be produced before it is able to trigger the next output event. As previously, its benefits include low computational resource consumption and speed.

Linear and cubic log-luminance reconstruction The log-luminance reconstruction method aims to recreate the log-luminance curves seen by the pixels in the target sensor size, then extrapolate the events produced by the average of these curves (see [7] for more details). The curves can be estimated with linear or cubic interpolation, both *linear* and *cubic* methods are considered in this paper.

In contrast to both previous spatial reduction methods, this log-luminance reconstruction needs information on when in the future will be the next event, which is obviously unknown in the current timestamps. Even though the

real-time processing needs to adjust the algorithm, the log-luminance reconstruction has the best optical coherence out of the existing event downscaling methods.

2.2.2 Spiking neural network-based pooling

As mentioned in [7], it is possible to elegantly use a Spiking Neural Network (SNN) [12] of Leaky-Integrate-and-Fire (LIF) neurons [8] to downscale events, using a simple 2-layer network. An input 2D layer of size $width \times height$ is connected to a smaller 2D layer of size $(width/ratio) \times (height/ratio)$, which implements a convolutional layer with a kernel size $ratio \times ratio$, a stride $ratio$ and no padding. Two different versions were implemented to downscale events using SNNs, depending on whether the positive and negative events are handled separately or not. These two versions will respectively be referenced as *separate SNN* and *mutual SNN* methods in the remainder. Furthermore, in order to trigger downscaled events that include both polarities, the input and output SNN layers are actually composed of $2 \times (width \times height)$ neurons (one *feature map* per polarity).

The run time of the SNN spatial downscaling method suffers from the limitations of the SpiNN-3 board [3] it is run on. The number of input events per simulation time-step does not influence the simulation run time, contrary to the number of neurons and connections, i.e. to the spatial dividing factor.

The six spatial event downscaling methods described above were experimentally evaluated in this work at four different dividing factors: 2 (resolution of 64×64), 4 (resolution of 32×32), 8 (resolution of 16×16) and 10 (resolution of 13×13).

2.2.3 Temporal event downscaling

In the present work, we introduce four temporal downscaling methods, reducing the data flow by sub-sampling events without impacting the spatial resolution. These methods are adapted to scenarios that highly differ from the ones where a spatial downscaling method would be used. We believe their use is especially relevant for embedded models, whose computing power may not be sufficient to record and process all events at the time of their arrival, but which do not necessarily have a limit on spatial resolution.

Temporal funnelling The temporal funnelling method comes close to *Tonic*'s temporal "Downsample" method [9]. Similarly to the spatial funnelling technique we implemented (see Paragraph 2.2.1), it involves the pooling of events according to their timestamp rather than spatial coordinates and removes any duplicates. Event data is expressed with a μs temporal precision. Pooling them according to their timestamp amounts to keeping only one event of the current polarity at the current pixel in the time window defined by the temporal factor t, which leads to changing the temporal precision.

We differentiate between two types of temporal funnelling:

- asynchronous temporal funnelling: each event kept per pixel, per polarity and per time-window maintains its original timestamp, with a thinner temporal grain than the one defined by t.
- synchronous temporal funnelling: each of those events is kept but with a timestamp rounded to the following t. This aims to illustrate the behaviour of an embedded system which would accumulate the kept events into frames, once all the events of the corresponding time window are processed.

The main disadvantages of these methods are the induced loss of precision and the increased event density. As for the spatial funnelling, this amounts to simply updating the memory address of the events' t coordinates, thus having a complexity $O(n)$. These two temporal funnelling downscaling methods were experimentally evaluated in this work at five different dividing factors: with time-window lengths of $0.05s$, $0.1s$ and $0.5s$ for the synchronous funnelling, and $1s$ and $5s$ for the asynchronous funnelling. These were agreed upon as it was immediately obvious to us that humans would understand the data obtained with the asynchronous method much better and for a longer time window, while they would have a harder time understanding events accumulated into frames displayed at the end of similar time windows. We thus decided to differentiate the factors used between the synchronous and asynchronous methods in order to limit the experimental time and not to collect expected and therefore irrelevant data.

Structural downscaling The structural downscaling methods can be subdivided into a deterministic one and a stochastic one. The deterministic method keeps every k^{th} event out of N. The stochastic method filters events with a probability p. This strategy has the benefit of maintaining the original time and space scales. It handles events efficiently, simply deciding whether to keep or discard each new event. It should be highlighted that this strategy simulates the real-world situation where an embedded system is overflowed with a dense event stream and is not able to process them all: some events will just be dropped [4].

These two temporal structural downscaling methods were experimentally evaluated in this work at three different dividing factors p: 10%, 1% and 0.1%. In other words, the deterministic factor k was respectively set dividing 10, 100

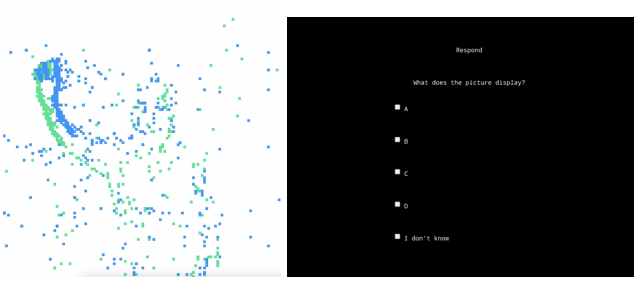

Figure 2. Interface presented to the subjects during the experiment.

Figure 3. Subjects performing the experiment.

and 1000 and k was converted into p according to Eq. 1.

$$p = 100 \times \frac{1}{k} \tag{1}$$

2.3. Interface

Participants interacted with the interface to classify different samples of the 4 gestures, with different downscaling techniques and resolutions recorded from 5 different users.

To capture the user input, the interface was implemented using OpenSesame [11], an open-source tool for experiment creation in the fields of psychology, neuroscience, and experimental economics. This interface allows for easy storage of the gesture shown in each sample, its downscaling technique and resolution, enabling also to capture the subject's response and its velocity.

The experiment encompassed a training phase to familiarise subjects with the four gestures, and a testing phase to record the data for analysis. During both phases, every sample was showcased for 10 seconds in mp4 format. Afterwards, subjects could click one of the 4 options (A, B, C, D) corresponding to the gestures or otherwise click "I don't know" (see Fig. 2).

In the training phase, subjects were presented with the different gestures without any downscaling and received a short feedback on the correctness of their answer. This training was repeated twice per participant. Afterwards, in the testing phase, the samples were shown without any feedback. All these visualisations were preceded by a brief experiment explanation, as well as demographic questions, such as gender and age range. Moreover, during the whole experiment, subjects had access to a poster (see Fig. 1) with illustrations showing each gesture involved in the study.

2.4. Protocol

The study involved two experiments, one to evaluate the spatial techniques and another to evaluate the temporal techniques. Both were performed by 30 subjects each. The participants gave informed consent to participate and were informed of the purpose and nature of the study. The study was conducted in designated rooms intended only for participants, on laptops with screens of similar size and quality. The participants were invited into the room at different intervals where they could choose one of the 3 stations which positioning prevented access to the answers of other participants (see Fig. 3).

In the experiment to assess spatial methods, participants identified 122 samples, with an average of 8m30s in the total task. In the experiment to assess temporal methods, participants had to identify 44 samples, with an average of 3m06s in the total task. In both experiments, the expected response time per sample was 15 seconds. However, in most cases, the subjects were able to choose an answer faster (about 4,45 seconds).

To reduce selection bias, participants of different genders, across all ages took part in the experiment (see Tab. 1). To reduce expectation bias, participants were not informed of the techniques they were classifying at any moment. To minimise order bias, the order of the shown samples was randomised.

3. Experimental results

3.1. Assessing human performance

In this subsection, we assess the quality of human classification by analysing the accuracy, the number of unknown answers, the time to reply and the average number of events for the different downscaling methods.

As expected, the human accuracy decreases and the percentage of unknown answers increases as the dividing factor increments across all spatial downscaling methods (see Fig. 4). We can observe that separate SNN and spatial funnelling are the techniques with the best human accuracy while log-luminance techniques have the worst results.

Human accuracy drops below the chance level (25%) for

Age	18-25 y.o.	More than 25 years old	Total
Female	11	11	22
Male	20	17	37
Other	0	1	1
Total	31	29	**60**

Table 1. Demographic repartition of the experiment participants.

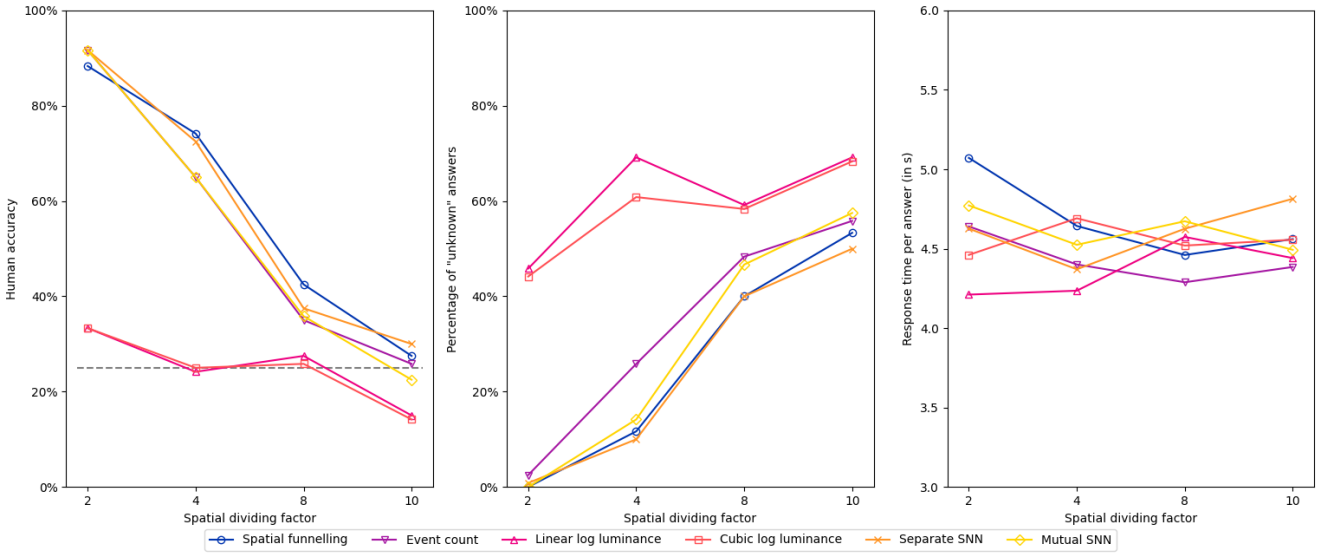

Figure 4. Human accuracy, rate of ”unknown” responses and human response time to event data downsized spatially.

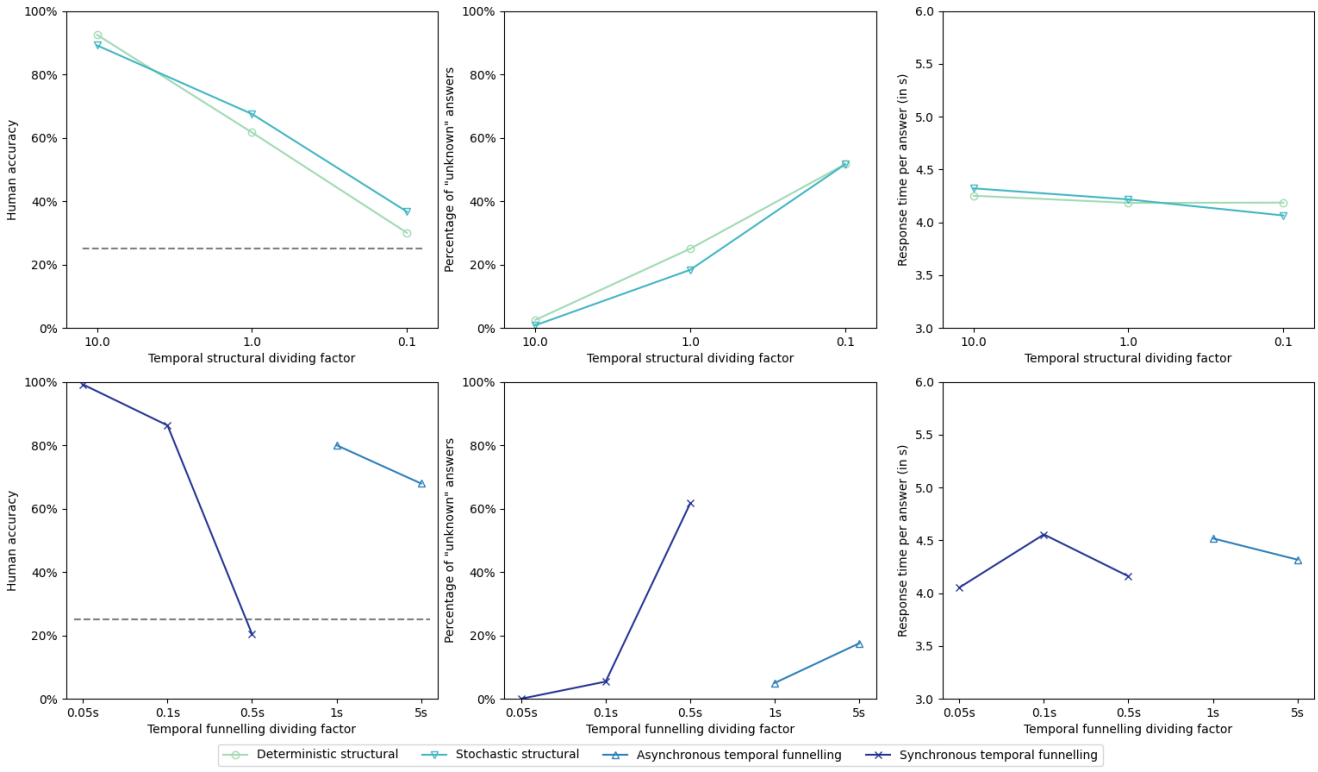

Figure 5. Human accuracy, rate of ”unknown” responses and human response time to event data downsized temporally.

a spatial diving factor 10 in the log-luminance and mutual SNN techniques. The biggest drop in human accuracy occurs from factor 4 to factor 8. Exceptionally, this fall is less steep for log-luminance techniques, where it even increases slightly but always stays close to the chance level threshold. The notable difference of human accuracy and percentage of unknown answers between techniques does not translate to the time per answer results, which are very homogeneous among techniques (see Fig. 4).

The temporal techniques with similar resolutions are assessed in Fig. 5 and in the following paragraph. Amongst the structural techniques, the stochastic outperforms the de-

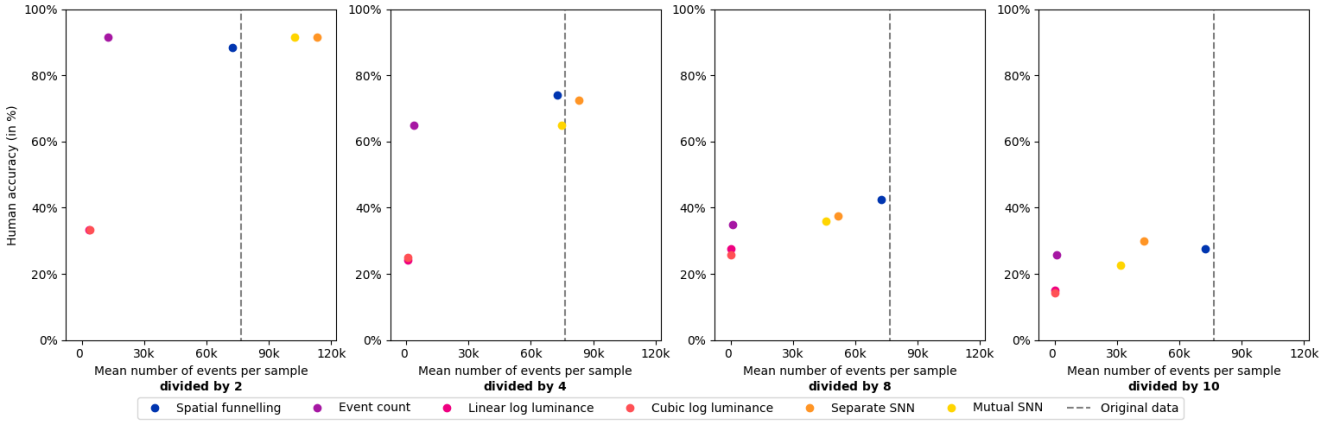

Figure 6. Human accuracy according to the number of events produced by spatial downscaling methods.

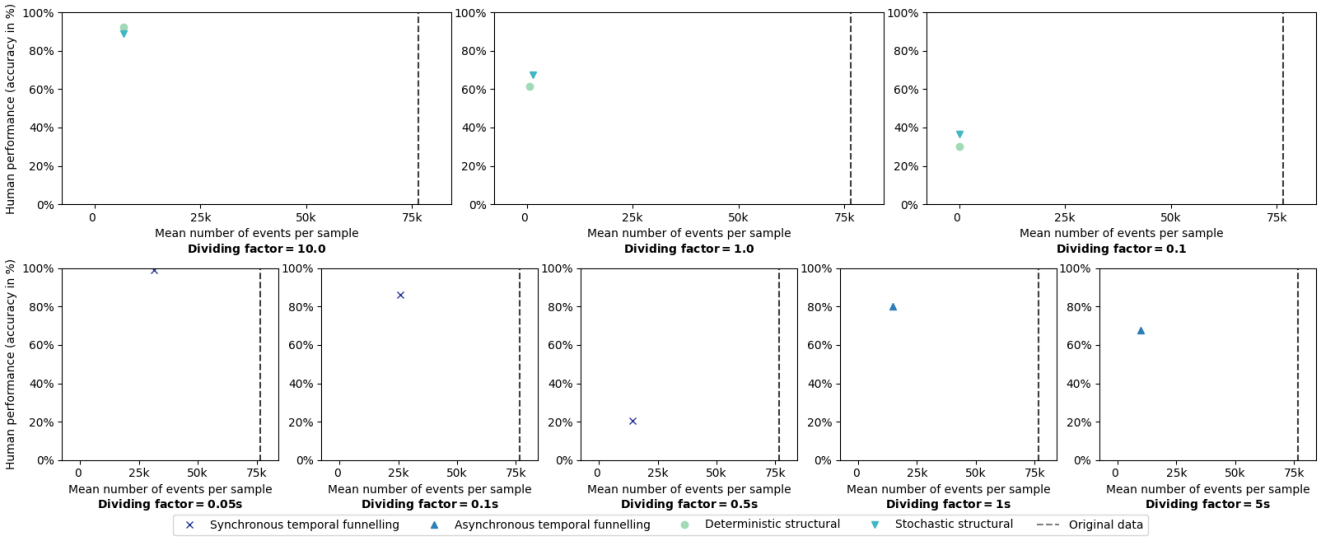

Figure 7. Human accuracy according to the number of events produced by temporal downscaling methods.

terministic. As for the temporal funnelling techniques, the asynchronous technique obtains better results, with a small percentage of unknown answers and a very notable accuracy of 70% despite an accumulation of events over $5s$. The temporal downscaling techniques have overall a better human accuracy and a lesser amount of unknown answers than the spatial techniques, which might be explained by the higher number of events. The time per answer is also faster than for spatial techniques. Only the synchronous temporal funnelling technique falls below the chance level when accumulated over a time window of $0.5s$.

Fig. 6 plots the human accuracy by spatial techniques against the number of events kept and clearly shows that the two techniques with the best overall human performance, Spatial Funnelling and Separate SNN, also have the highest number of events from dividing factor 4 onwards. Moreover, the techniques with the worst human accuracy, the log-luminance techniques, have the smallest number of events kept with respect to the original number of events.

Regarding the temporal downscaling techniques (see Fig. 7), we can observe that the number of events kept is the same across all dividing factors. As mentioned, the stochastic technique's accuracy exceeds the deterministic's performance.

3.2. Comparing human and machine performance

In this subsection, we assess the quality of human classification compared to the performance achieved by a neural network. We compare the results we experimentally collected with the results output by the SNN classifier introduced by Fang et al. in 2021, consisting of a new spiking neuron model called Parametric Leaky Integrate-and-Fire (PLIF) [2]. This model allows the authors to implement a backpropagation learning algorithm, applied to a classification task. They present the results obtained when classifying traditional RGB datasets, as well as neuromorphic

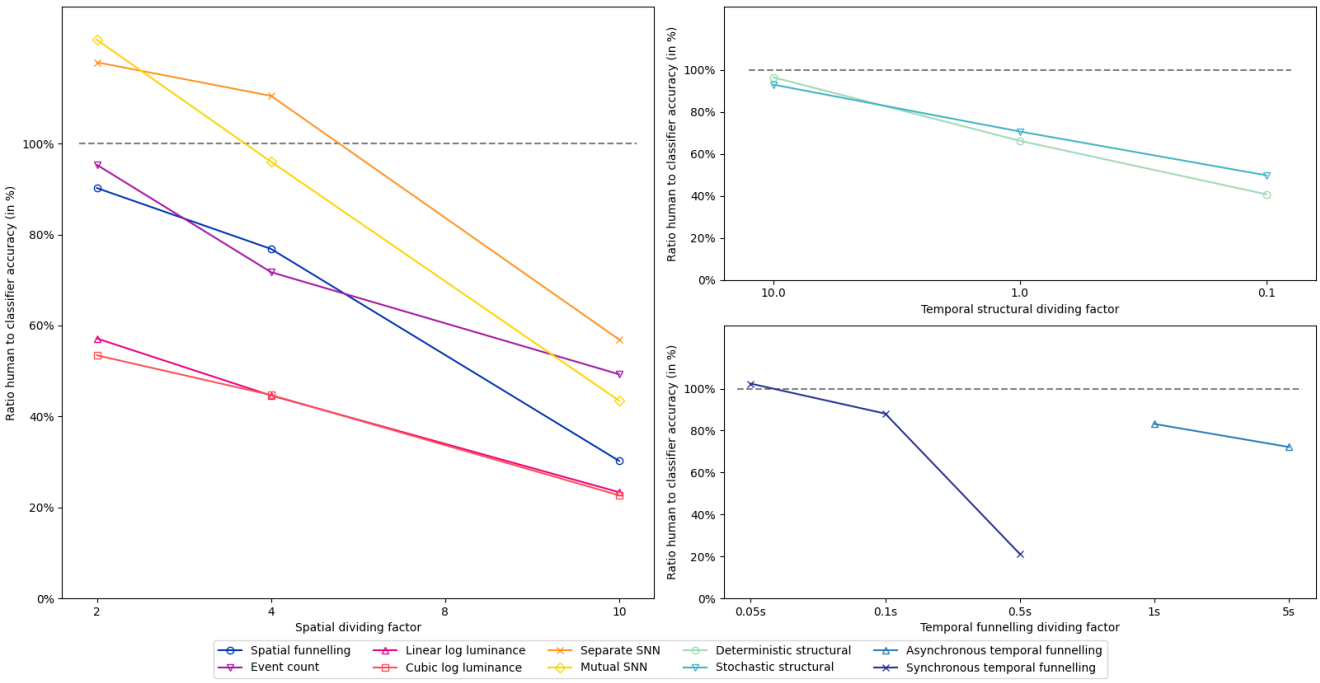

Figure 8. Ratio of the human accuracy to the classifier accuracy, for each downscale factor, for spatial and temporal. When the curve exceeds 100% (dotted grey line), the human performance is higher than the classifier's; when it falls below 100%, the classifier outperforms humans.

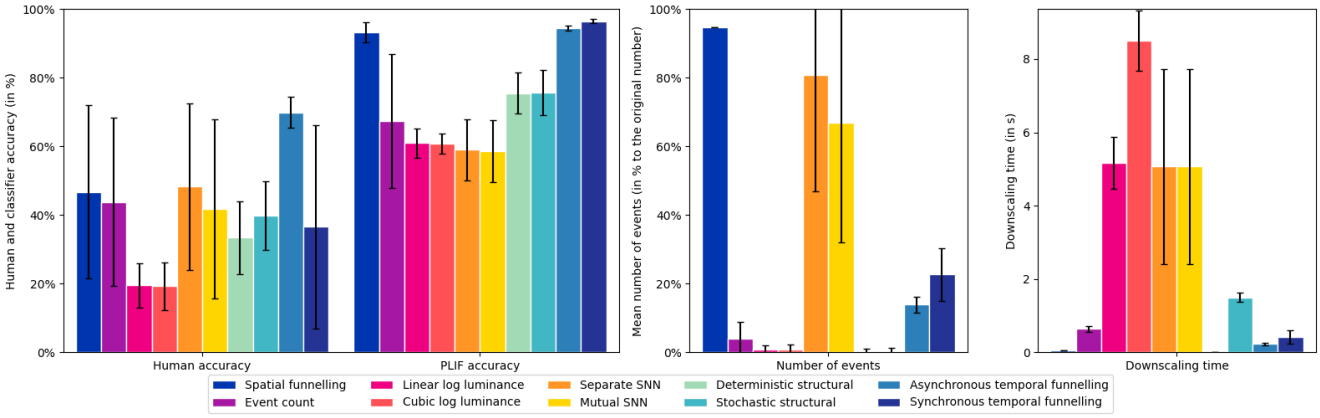

Figure 9. Global comparison between spatial and temporal event downscaling methods, according to the overall human and machine performance, number of events and downscaling time of $1s$ of input event data (in s). All those criteria are weighted by the downscaling factor: the bigger the reduction, the higher the weight of the corresponding value. The number of events corresponds to the ratio of reduced events to the original number (in %). Each barplot presents the weighted mean of the corresponding value, and the black error bars the weighted standard deviation.

datasets such as DVS128 Gesture [1]. It is important to note that the authors chose to process the datasets as frames, and this precisely because of their high number of events.

For the structural and temporal funnelling techniques, the classifier outperforms the human as expected, with a minor exception in the Synchronous Temporal Funnelling method with a time-window of $0.05s$ (see Fig. 8). The difference between human and machine performance is especially stark for the synchronous temporal funnelling: this underlines the bias of using a frame-based classifier with event data. For the spatial methods, we encounter more evident exceptions in the mutual and separate SNN at the dividing factors between 2 and 4.

Fig. 9 adds a fourth comparison factor to the three evaluated so far (human accuracy, machine performance and

number of events compared to the original data): the downscaling time. The best method would optimise human and machine accuracy while minimising the number of events and downscaling time. Amongst the spatial downscaling methods, we observe once again that the techniques with the best compromise seem to be Spatial Funnelling and SNN. Mutual SNN is slightly better because it achieves an equivalent human accuracy with a significantly lower number of events, even though the downscaling time increments significantly. Once again, log-luminance methods don't provide satisfactory results; even if they have an almost negligible number of events and are optically coherent with the behaviour of an event camera (see [7]), the accuracy is not good enough and the downscaling time is higher than expected.

Amongst the temporal structural downscaling methods, the deterministic technique has a very similar accuracy to stochastic, with a lot less time taken to downscale. Less evident, in temporal funnelling, synchronous offers better global results than the asynchronous technique, with a lower downscaling time and number of events traded off for a decrease in human accuracy.

The broad standard deviation observed in Fig. 9 on the human accuracy measured on five methods (spatial funnelling, event count, SNN separate and mutual and especially synchronous temporal funnelling) can be explained by the wide variation in human performance depending on the intensity of reduction (see Fig. 6 and 7). This reinforces the overall value of the asynchronous temporal funnelling method, whose standard deviation is less extensive although the performances are measured on significantly higher dividing factors.

4. Conclusion

We have presented a study on the downscaling of event data for gesture classification using human participants. Our study showed that a certain size threshold needs to be maintained to ensure that human performance does not fall below the chance level. This threshold is close to the factor 8 for spatial methods, 0.1 for temporal structural methods and $0.5s$ for temporal funnelling methods. Furthermore, our comparison of different downscaling methods revealed that the quality of the data obtained from these methods is not uniform. Finally, our results also highlight some discrepancies between human and machine learning approaches to gesture classification using event data. Human accuracy is higher than machine accuracy in specific dividing factors for the Synchronous Temporal Funnelling and Mutual and Separate SNN techniques. This study sheds light on the potential limitations of event data downscaling and provides insights into the human perception of gesture classification using event data. The findings of this study have implications for the design and implementation of embedded computer vision systems that rely on event data, and may also inform the development of more accurate machine learning algorithms for gesture recognition.

Acknowledgements

This work was supported by the European Union's ERA-NET CHIST-ERA 2018 research and innovation programme under grant agreement ANR-19-CHR3-0008.

The authors are grateful to the OPAL infrastructure from Université Côte d'Azur for providing resources and support.

References

[1] A. Amir et al. A Low Power, Fully Event-Based Gesture Recognition System. In *CVPR*. IEEE, 2017. 1, 2, 7

[2] Wei Fang et al. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *ICCV*, 2021. 6

[3] Steve Furber and Petrut Bogdan. *Spinnaker - a spiking neural network architecture*. NOW Publishers INC, 2020. 3

[4] G. Gallego, T. Delbruck, G. Orchard, and al. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 1, 3

[5] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, nov 2020. 1

[6] Amélie Gruel, Jean Martinet, Bernabé Linares-Barranco, and Teresa Serrano-Gotarredona. Performance comparison of dvs data spatial downscaling methods using spiking neural networks. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 6483–6491, 2023. 1, 2

[7] Amélie Gruel, Jean Martinet, Teresa Serrano-Gotarredona, and Bernabé Linares-Barranco. Event data downscaling for embedded computer vision. In *VISAPP*, 2022. 1, 2, 3, 8

[8] Eugene M. Izhikevich. Which model to use for cortical spiking neurons? *IEEE transactions on neural networks*, 15(5):1063–1070, Sep 2004. 3

[9] Gregor Lenz, Kenneth Chaney, Sumit Bam Shrestha, Omar Oubari, Serge Picaud, and Guido Zarrella. Tonic: event-based datasets and transformations., jul 2021. Documentation available under https://tonic.readthedocs.io. 3

[10] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128× 128 120 db 15 μs latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, 43(2):566–576, 2008. 1

[11] Sebastiaan Mathôt, Daniel Schreij, and Jan Theeuwes. Opensesame: An open-source, graphical experiment builder for the social sciences. In *Behavior Research Methods 44(2)*, pages 314–324, 2012. 4

[12] Hélène Paugam-Moisy and Sander M. Bohte. Computing with Spiking Neuron Networks. In *Handbook of Natural Computing*. Springer-Verlag, Sept. 2012. 3

[13] Megha Srivastava, Tatsunori B. Hashimoto, and Percy Liang. Robustness to spurious correlations via human annotations. *CoRR*, abs/2007.06661, 2020. 1