ARTICLES
https://doi.org/10.1038/s42256-019-0110-8
nature machine intelligence

Human action recognition with a large-scale brain-inspired photonic computer

Piotr Antonik[1*], Nicolas Marsal[1], Daniel Brunner[2] and Damien Rontani[1*]

The recognition of human actions in video streams is a challenging task in computer vision, with cardinal applications in brain–computer interfaces and surveillance, for example. Recently, deep learning has produced remarkable results, but it can be hard to use in practice, as its training requires large datasets and special-purpose and energy-consuming hardware. In this work, we propose a photonic hardware approach. Our experimental set-up comprises off-the-shelf components and implements an easy-to-train recurrent neural network with 16,384 nodes, scalable to hundreds of thousands of nodes. The system, based on the reservoir computing paradigm, is trained to recognize six human actions from the KTH video database using either raw frames as inputs or a set of features extracted with the histograms of an oriented gradients algorithm. We report a classification accuracy of 91.3%, comparable to state-of-the-art digital implementations, while promising a higher processing speed in comparison to the existing hardware approaches. Because of the massively parallel processing capabilities offered by photonic architectures, we anticipate that this work will pave the way towards simply reconfigurable and energy-efficient solutions for real-time video processing.

In recent years, human action recognition has become one of the most popular research areas in the field of computer vision[1]. The driving force behind this development is the range of potential applications in areas such as surveillance, control and analysis[2]. Surveillance is concerned with tracking one or several subjects over time and detecting specific actions. A typical example is the surveillance of a parking lot for the prevention of car theft. Applications related to system control make use of captured motions to provide control functionality in games, virtual environments or to control remote devices[3], and the detailed automatic analysis of motion could be used in clinical studies of orthopaedic patients or to help athletes improve their performance[2].

The recognition of human activity from video sequences is challenging. There are numerous problems to be overcome, such as background clutter, partial occlusion, changes in scale or viewpoint, lighting and appearance[4]. Deep learning has been successfully applied to speech recognition, natural language processing and recommendation systems, and has recently been introduced into video-based human action recognition research[1]. The many advantages of these hierarchical approaches—raw video inputs, automatically deduced features and recognition of complex actions—have attracted much interest from the research community. However, these approaches also have several drawbacks, such as the need for (very) large datasets, the non-trivial tuning of hyperparameters and the time- and energy-consuming training process, which commonly requires dedicated high-end hardware such as graphical processing units.

In this work, we propose an optical signal processing system for the classification of video-based human actions. The idea of optical computing has been investigated for decades, because photons essentially propagate without generating heat and suffering from signal degradation due to inductive and capacitive effects, and thus promise a high level of parallelism in, for example, optical signal transmission. Neural networks would benefit greatly from parallel signal transmission, which, as shown by the increasing usage of optical interconnects in modern computing systems, is one of the strong features of photonics. An optical approach could thus allow us to build high-speed and energy-efficient photonic computing devices.

Our experimental optical system implements a shallow recurrent neural network under the so-called reservoir computing (RC) paradigm. RC is a set of machine learning methods for designing and training artificial neural networks[5,6]. The idea behind these techniques is to exploit the dynamics of a random recurrent neural network to process time series by only training a linear output layer. The resulting system is significantly easier to train—instead of the entire network, only the readout layer is optimized by solving a system of linear equations[7]. Furthermore, as fewer parameters are inferred during training, the network can be trained on significantly smaller datasets without the risk of overfitting. The performance of the numerous experimental implementations of RC in electronics[8], optoelectronics[9–12], optics[13–16] and integrated on chip technology[17] is comparable to other digital algorithms on a series of benchmark tasks, such as wireless channel equalization[5], phoneme recognition[18] and prediction of future evolution of financial[19] and chaotic[20] time series. Finally, it has been shown that the readout layer of photonic reservoir computers can be implemented optically and trained using a digital micromirror device[21].

In this Article, we present an optoelectronic reservoir computer, inspired by refs. [21,22]. The system is based on phase modulation of a spatially extended planar wave by means of a spatial light modulator (SLM). Our scheme offers notable parallelization potential through simultaneous optical processing of the nodes of the reservoir computer, while the physical resolution of the SLM defines the maximal network size. This allows for a significantly increased scalability of the network, which is vital for successfully solving the challenging tasks in computer vision. The experimental set-up can accommodate a reservoir of 16,384 nodes, while the physical limitation of the concept is set to be as high as 262,144 neurons. The input and output layers, as well as the recurrence of the network, are realized digitally in this work.

[1]LMOPS EA 4423 Laboratory, CentraleSupélec & Université de Lorraine, Metz, France. [2]FEMTO-ST Institute/Optics Department, CNRS & Université Bourgogne Franche-Comté, Besançon, France. *e-mail: piotr.antonik@centralesupelec.fr; damien.rontani@centralesupelec.fr

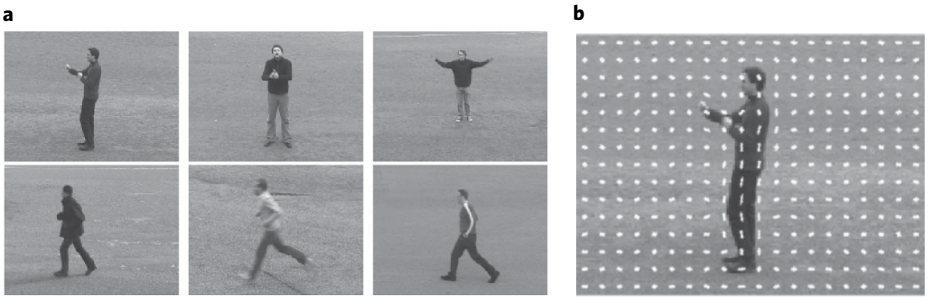

Fig. 1 | Examples of KTH frames and HOG feaures. a, Examples of action frames from the KTH database. Top row, left to right: boxing, hand clapping, hand waving. Bottom row, left to right: jogging, running and walking. Six different subjects are illustrated out of a total of 25. All videos were taken outdoors over a homogeneous background, which corresponds to the 's1' subset of the full database. **b**, Example of HOG features computed in Matlab for a frame of the KTH dataset. The HOG features are visualized using a grid of rose plots. The grid dimensions (20×15 here) are determined by the ratio between the image and cell sizes. Each rose plot shows the distribution of gradient orientations within a HOG cell. The length of each petal of the rose is proportional to the contribution of each orientation within the histogram. The plot thus displays the edge directions, which are normal to the gradient directions. In this example, it allows the pose of the subject to be captured.

The system is benchmarked on the popular KTH database[23], which contains video recordings of six different motions (walking, jogging, running, boxing, hand waving and hand clapping) performed by 25 subjects (Fig. 1). At the pre-processing stage, the histograms of oriented gradients (HOG) algorithm[24] (described later) is used to extract spatial and shape information from individual video frames. The photonic reservoir computer is used to classify the six motions given the resulting HOG features.

The set-up is evaluated both experimentally and in simulations. The numerical model was designed to mimic the experiment as accurately as possible. It is based on the same nonlinearity, trained and tested on the same data, and the hyperparameters are optimized in the same way. We investigate the scalability of our approach with network sizes ranging from 1,024 to 16,384 nodes and report classification accuracy as high as 92%, which is comparable to the state-of-the-art rates of 90.7–95.6% achieved with far more complex and demanding architectures implemented on noiseless digital processors[1]. This work thus shows that a challenging computer vision task can be efficiently solved with a simple photonic system. It represents a successfull first step towards a video processing system with electronic pre-processing stage (HOG features) and a fully optical reservoir computer, which benefits from the intrinsic parallelism of photonics and thus offers a highly scalable and potentially energy-efficient neural network.

Results

Before presenting the results of this study, we introduce the video-based human action classification task in the context of RC, and then present the experimental set-up. The theory of RC is provided in the Methods.

Classification of human action with a reservoir computer. The principles of the human action recognition task in the context of RC are illustrated in Fig. 2. In this work, we use the popular KTH database of human actions[23], publicly available online, which consists of video recordings of six different motions (walking, jogging, running, boxing, hand waving and hand clapping) performed by 25 subjects. In particular, we focus on the first scenario 's1', containing outdoor videos shot over a uniform background (illustrated in Fig. 1). Each subject performs each motion four times, which results in a dataset of 600 video sequences of variable lengths, ranging from 24 to 239 frames. More details on the video properties of the dataset are provided in the Methods. All videos are concatenated together and split into individual frames, giving the raw video stream (Fig. 2a), carried forward to the pre-processing stage (Fig. 2b).

Feature extraction is a common approach in computer vision for providing the classification system—in this case a photonic reservoir computer—with the most relevant information. We tested our reservoir computer with raw frames, but the classification errors were significantly higher than the state of the art. We thus turned to the HOG algorithm, introduced by Dalal and Triggs[24]. With this, an intra-frame spatial gradient is computed for each pixel and then pooled into one common gradient histogram. Such HOG features are widely used in computer vision and image processing with the intention of aiding the localization and detection of objects (see, for example, ref. [25]). This method is particularly well suited for pedestrian detection (as well as a variety of common animals and vehicles) in static imagery. The main idea is that the local object appearance and shape can often be expressed well enough by a distribution of local intensity gradients or edges' directions. The HOG algorithm is further discussed in the Methods. To reduce the number of resulting HOG features and simplify computations, we applied principal component analysis[26,27] based on the covariance method[28]. We chose to keep the first 2,000 components (out of 9,576), whose eigenvalues account for 91.6% of the total variability in the data.

The training of the reservoir computer, illustrated in Fig. 2c, was performed framewise on a subset of 450 video sequences, each containing a single motion sequence. A total of 150 sequences were used to evaluate the performance of the system. Figure 2d illustrates the six binary classifiers, introduced to distinguish the motions: six output nodes have been trained to give a '1' for each frame of the correct motion and '0' for the other frames. The winner-takes-all approach, shown in Fig. 2e, is used to classify each individual frame. The classifier output is evaluated throughout the full video sequence (from the first frame to the last) and the final result corresponds to the class having the majority of frames within the sequence attributed to it.

During training, the normalized mean square error (NMSE) cost function (equation (3)) was used to minimize the error between the reservoir output and the target class. In this study, the final classification did not require the output of the correct class to be as close as possible to '1', with the others close to '0'. Because we use the winner-takes-all approach, all it takes for the correct class to 'win' is to be slightly higher than the others. In other words, a lower NMSE does not necessarily mean fewer classification errors. We therefore used a different error metric based on the confusion matrix[23]. Here, the confusion matrix is a 6×6 array (dictated by the number of classes) computed for the entire video stream, each cell $p(i, j)$ giving the percentage of actions of class i

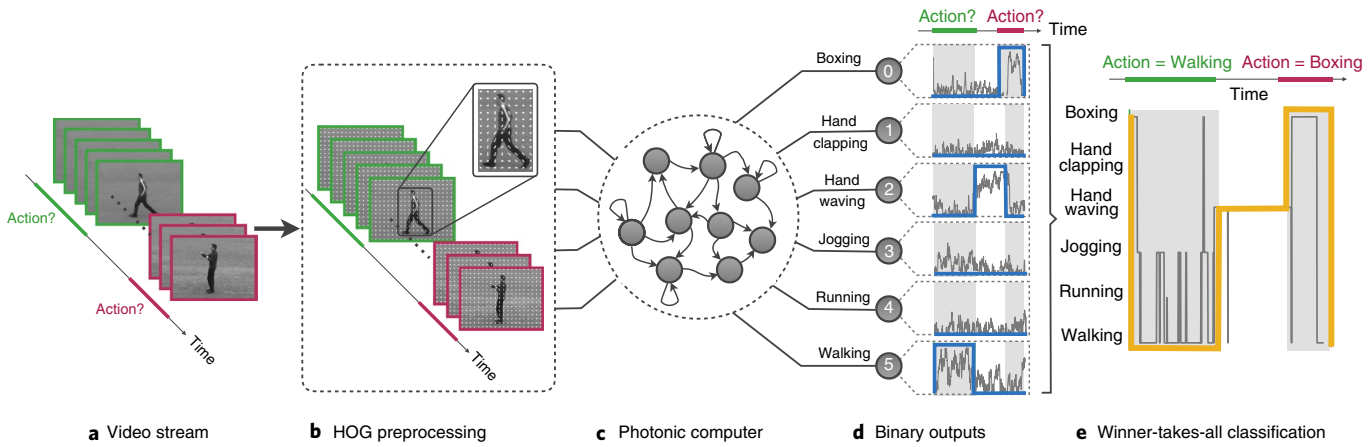

Fig. 2 | Scheme of the principle of how our reservoir computer solves the human action classification task. a, The video input stream is a concatenation of the 600 video sequences available in the KTH dataset; 450 sequences were used for training and 150 for testing. **b**, The input stream undergoes a pre-processing stage, where the HOG algorithm is applied to each individual frame. The dimensionality reduction through principal component analysis is not illustrated in this figure. **c**, Selected features are fed into the photonic reservoir computer, which is trained to classify each individual frame. **d**, This is achieved by defining six binary output nodes, one for each action class, which are trained to output 1 for a frame of the corresponding class and 0 for the others. Target outputs are shown in blue. **e**, The framewise classification is obtained by selecting the node with the maximum output, that is, the winner-takes-all approach. The final decision for a video sequence is given by the class attributed to the most frames of the sequence. The target class is shown in yellow. Two examples illustrate the entire process. A boxing sequence, highlighted in red in **a** and **b**, is classified unambiguously in **e**, as all output nodes in **d** remain low except for the one corresponding to boxing, which generates a clear spike. A walking sequence, highlighted in green, is more uncertain, as two output nodes—jogging and walking—generate high responses in **d**. Therefore, the reservoir output (**e**) oscillates between the two classes (the faint vertical lines in the light grey left-hand side region). However, because more frames in the sequence are classified as walking (74.5%) than jogging (23.6%), the entire sequence is correctly classified as walking.

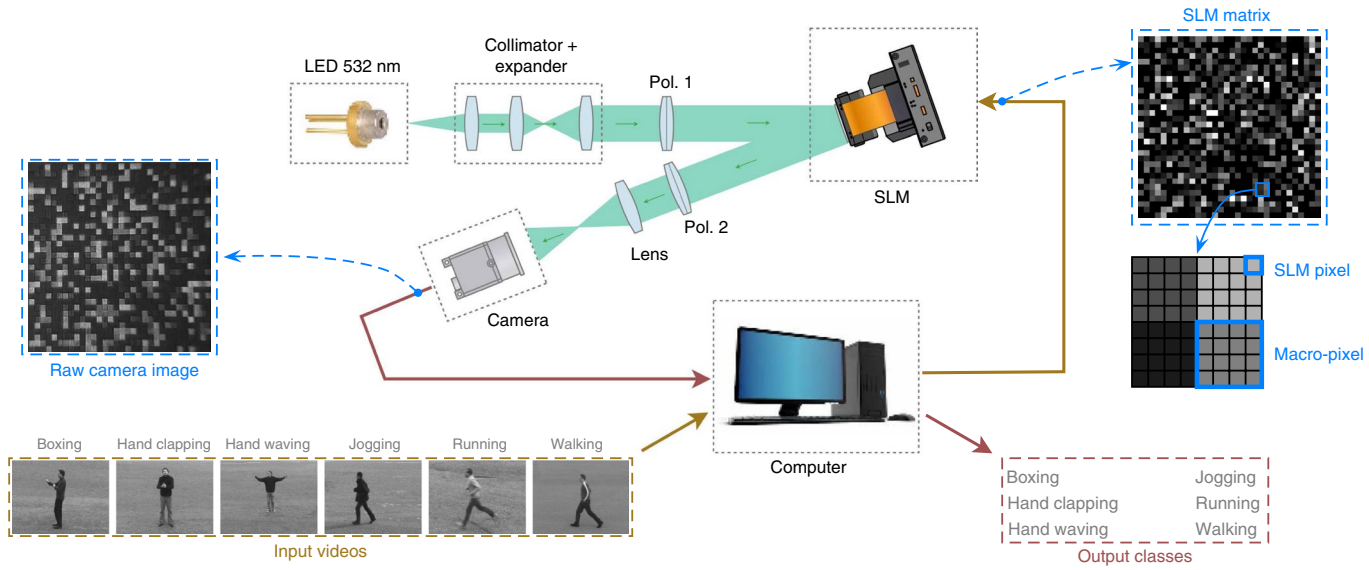

Fig. 3 | Illustration of the experimental set-up, composed of an optical arm, connected to a computer. The output of a green LED (532 nm) is collimated and expanded (collimator + expander), then polarized (Pol. 1) and used to illuminate the surface of the SLM. The latter is being imaged by a high-speed camera through a second polarizer (Pol. 2) and an imaging lens. Both the camera and the SLM are controlled by a computer, running a Matlab script. The latter generates the inputs from the input videos, and computes the values of pixels to be loaded on the SLM (the SLM matrix). Groups of small individual pixels of the SLM are combined into larger macro-pixels, which are easier to separate on the raw camera image. The computer uses the data from the camera to extract the reservoir states, compute the outputs and generate the output classes.

classified into class j. In other words, the diagonal of the confusion matrix contains the correct classification produced by the system, while non-zero elements off the diagonal correspond to errors. We use the confusion matrix to compute a new metric for the reservoir computer performance, called the score, given by the sum of the diagonal elements. A perfect classification corresponds to a score of 600, as all the six actions have been recognized with 100% accuracy.

Photonic reservoir computer. A typical discrete-time reservoir computer contains a large number N of internal variables $x_{i\in 0\ldots N-1}(n)$ evolving in discrete time $n \in \mathrm{Z}$, as given by

$$x_i(n+1) = f_{\mathrm{NL},I}\left(\sum_{j=0}^{N-1} W_{ij}^{\mathrm{res}} x_j(n) + \sum_{j=0}^{K-1} B_{ij} u_j(n)\right) \quad (1)$$

where $f_{\mathrm{NL},I}$ is a nonlinear function (in this work, $f_{\mathrm{NL},I}(x) = \lfloor I_0 \sin^2(\lfloor x \rfloor_8) \rfloor_{10}$, where $\lfloor \ldots \rfloor_n$ is an n-bit quantization operation), W_{ij}^{res} is an $N \times N$ matrix of interconnecting weights between the neurons of the neural network, $u_j(n)$ is an input with K dimensions, and B_{ij} is an $N \times K$ matrix of input weights, often referred to as the input mask. Further information on the principles of RC and the properties of B_{ij} and W_{ij}^{res} is provided in the Methods.

Our experimental set-up implements equation (1) and is schematized in Fig. 3. It is composed of two parts: a free-space optical arm and a computer. The optical part implements the nonlinearity $f(x) = \sin^2(x)$ in equation (1). It is powered by a green light-emitting diode (LED) source at 532 nm (Thorlabs M530L3) set to a power level of 10.5 mW. The choice of wavelength was based on the availability of optical components and the ease of use and calibration of the set-up in the visible spectrum, and the optical power was adjusted to provide sufficient illumination of the SLM to generate the highest contrast, with adequate exposure settings on the camera. The output beam was linearly polarized, collimated and expanded to roughly 17 mm in diameter to evenly illuminate the entire 7.68 mm × 7.68 mm surface of the SLM (Meadowlark XY Phase P512–0532 with 8 bit resolution). In simplified terms, a SLM can be operated as a variable, spatially resolved wave plate, and its index of refraction along the slow axis can be controlled electronically. Accordingly, a linearly polarized illumination beam, parallel to the slow axis of the SLM, would be reflected with a phase-only modulation. If a beam were parallel to the fast axis instead, one would observe no modulation by the electronic control signal. In this set-up, an illumination beam oriented at 45° with respect to the slow axis provided equal optical field components to the fast and slow axes of the SLM. After reflection, the former remained unchanged, while the latter underwent phase modulation. A second polarizer transformed the phase difference between the two components into intensity modulations, which were in turn imaged onto a high-speed camera (Allied Vision Mako U-130B with 10 bit resolution). The imaging system was optimized for a compromise between imaging resolution and the field of view's extent.

The computer was used to run a Matlab script controlling both the SLM and the camera, taking care of loading the data into the SLM and obtaining images from the camera. The input mask B_{ij} and the interconnection matrix $\sum W_{ij}^{\mathrm{res}} x_j(n) + \sum B_{ij} u_j(n)$ were generated randomly at the beginning of the experiment. At each discrete timestep n, the input to the nonlinear function $\sum W_{ij}^{\mathrm{res}} x_j(n) + \sum B_{ij} u_j(n)$ was computed, and the resulting matrix was loaded onto the SLM device. The camera then recorded a picture of the SLM through the imaging lens and the polarizing optics. The raw image was cropped to the area of interest (the surface of the SLM) and averaged over the macro-pixels (see below), resulting in a square matrix that represents the updated reservoir states, given by equation (1). The states were rearranged into a vector $\mathbf{x}_j$ and used to compute the SLM's next input matrix at timestep $n+1$.

In this experiment, the reservoir size is defined by several factors. The device used here had a resolution of 512 × 512 pixels and allowed, in theory, for a network size of 512 × 512 = 262,144 neurons, if each individual pixel was used as a node. However, in our experiment this was challenging, because the SLM surface was slightly tilted with respect to the camera sensor. Consequently, only a limited region of the SLM was seen in focus by the camera, while the rest was blurry. Therefore, in this experiment, we only used the central 384 × 384 region of the SLM, and assigned square groups of pixels—macro-pixels—to individual reservoir nodes. For example, a small network of $N = 1{,}024$ nodes was obtained by setting the macro-pixel size to 12 × 12, while a large network ($N = 16{,}384$) was obtained by reducing the macro-pixel size to 3 × 3 pixels on the SLM.

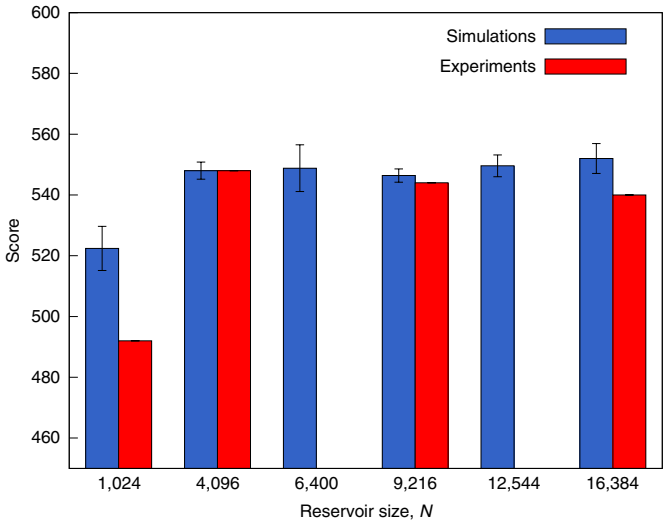

Fig. 4 | Performance of our photonic neuro-inspired architecture on the human action classification task. Different reservoir sizes have been investigated numerically (blue) and experimentally (red). The error bars on the numerical results show the score variability (s.d.) with five different input masks. Experimental variability could not be measured because of the long run time of the experiment.

The speed of the set-up is imposed by Matlab, that is, by the time needed to compute the next SLM matrix from the raw camera image. For a large reservoir of $N = 16{,}384$ nodes, our system was capable of processing two video frames per second. In the case of a small reservoir ($N = 1{,}024$), the matrix multiplication $\sum W_{ij}^{\mathrm{res}} x_j(n)$ (equation (1)) required fewer computations, and the processing speed was increased up to seven frames per second. The use of Matlab at this stage was a deliberate choice, as it lends considerable flexibility to the set-up, for example testing different pre-processing techniques, reservoir topologies and output decision-making layers by simply changing the code, that is without reconfiguration of the optical set-up. The system's speed limitation could be alleviated by replacing the computer with a dedicated digital signal processing (DSP) board or a field-programmable gate array (FPGA) chip, capable of performing the matrix-products computations in real time (as in, for example, ref. [29]). More importantly, the matrix could equally be offloaded to fully parallel optics[21,30]. As our SLM model supports refresh rates up to 300 Hz in overdrive mode, the hardware would be capable of processing a video stream in real time. Furthermore and because of its high frequency of operation, we could also theoretically time-multiplex up to 12 video streams at 25 f.p.s.

Reservoir size and classification performance. To test the potential of our large-scale architecture on a challenging computer-vision task, we studied the impact of the reservoir size on the classification performance. We investigated reservoir sizes from $N = 1{,}024$ up to $N = 16{,}384$, both numerically and experimentally. Figure 4 shows the resulting performance for different reservoir sizes in terms of the classification score (introduced in section 'Classification of human action with a reservoir computer'), computed during the testing phase. The hyperparameters were optimized from scratch for each reservoir size, and independently in simulations and experiments. Details on the optimization approach are provided in section 'The KTH dataset'. In numerical simulations, we investigated the performance with different random input and interconnection weights. We performed five distinctive simulations for each reservoir size

Table 1 | Performance of various state-of-the-art digital approaches compared to our best experimental result

					Performance	
Authors	**Method**	**Database split**	**Training time**	**Processing speed (f.p.s.)**	**s1 scenario (%)**	**Full database (%)**
Yadav et al.[38]	IP + SVM	80%–20%	–	–	–	98.20
Shi et al.[39]	DTD, DNN	9–16	–	–	–	95.6
Kovashka et al.[40]	BoW + SVM	8-8-9	–	–	–	94.53
Gilbert et al.[33]	HCF + SVM	LOOCV	~5.6 h	24	–	94.5
Baccouche et al.[41]	CNN & RNN	16–9	–	–	–	94.39
Ali and Wang[42]	DBN & SVM	50%–20%–30%	–	–	–	94.3
Wang et al.[43]	DT + SVM	16–9	–	–	–	94.2
Liu et al.[44]	MMI + SVM	LOOCV	–	–	–	94.15
Sun et al.[45]	FT + SVM	Auto	–	–	–	94.0
Veeriah et al.[46]	Differential RNN	16–9	–	–	–	93.96
Shu et al.[47]	SNN	9–16	–	–	95.3	92.3
Laptev et al.[48]	FT + SVM	8–8–9	–	–	–	91.8
Jhuang[31]	StC_2 + SVM	16–9	–	0.4	96.0	91.6
Klaeser et al.[49]	3D Grad + SVM	8–8–9	–	–	–	91.4
This work	**Photonic RC**	**75%–25%**	**1.6–5.5 h**	**2–7**	**91.3**	–
Grushin et al.[32]	LSTM	16–9	1 day	12–15	–	90.7
Ji et al.[50]	3DCNN	8–8–9	–	–	–	90.02
Escobar et al.[51]	MT cells	16–9	–	–	74.63	–
Schuldt et al.[23]	FT + SVM	8–8–9	–	–	–	71.83

'Database split' indicates how the KTH database was split for training and testing of the system. Most studies choose to split by the number of subjects into either two groups (training and test, for example 16 subjects for training, 9 for the test) or three groups (training, validation and test, for example 8–8–9). LOOCV corresponds to 'leave-one-out cross validation': the system is trained on 24 subjects and tested on the remaining one. Training times and processing speeds are not discussed in most of the works, focusing on the classification performance. Some studies report specific results on the s1 scenario, as considered in this work. BoW, bag of words; CNN, convolutional neural network; DBN, deep belief network; DNN, deep neural network; DT, dense trajectories; DTD, deep trajectory descriptor; FT, features; HCF, hierarchical compound features; IP, interest points; LSTM, long short-term memory neural network; MMI, maximization of mutual information; MT, middle temporal area of the visual cortex; RNN, recurrent neural network; SNN, spiking neural network; StC_2, space-time oriented C_2 features; SVM, support vector machine.

with different random weights and full optimization of the hyperparameters. Blue bars display the average performance and the error bars show the standard deviations, that is, the variability of the score due to different random weights. In the experiment (red bars) such statistical analysis was hampered by the long measurement duration (from a few days for smaller reservoirs up to a week for $N=16{,}384$). Such long experimental run times are due to the optimization of the hyperparameters through grid search (see section 'The KTH dataset'). With one set of hyperparameters, the experiment processes the full dataset (both the training and test stages) in 1.6–5.5 h, depending on the reservoir size.

The graph shows a steep increase in performance from a small reservoir size of $N=1{,}024$ nodes up to $N=4{,}096$, both in numerical simulations and the experiment. The average score at $N=4{,}096$ is 548 in both cases. The numerical results keep improving for large reservoirs, reaching an average score of 552 at $N=16{,}384$. The experimental results, on the other hand, exhibit a slight decrease in performance with large reservoirs. This downturn is due to experimental imperfections, such as tilt and misalignment of the area of interest cropped from raw camera images, which become more noticeable as the macro-pixels shrink. However, the decrease has little significance, with a 1.3% performance drop between $N=4{,}096$ and $N=16{,}384$.

Table 1 compares the performance of our optical experiment with state-of-the-art digital approaches (details are available in the listed papers). The table reports how the systems were trained on the KTH dataset (the database split), as well the training time and processing speed, wherever possible (those two metrics are very seldom reported in the literature, hence the large number of empty cells in the table). A few studies also report the system performance specifically on the s1 scenario, and are thus directly comparable to our results. In terms of performance, the photonic RC is short by 4.7% from the best results for the s1 scenario[31], but outperforms the support vector machine (SVM) approach in terms of processing speed by a factor of ten. The training time of our system is significantly shorter than deep approaches[32], and comparable to the SVM method with hierarchical compound features[33]. Our photonic approach thus offers a functional, more flexible and easy to train classification system. Furthermore, the recent development of integrated photonic reservoir computers[17] could give rise to very energy-efficient optical processors.

Figure 5 displays the confusion matrices for the best scores obtained numerically (552 at $N=16{,}384$) and experimentally (548 at $N=4{,}096$). The results obtained from the experiment agree very well with the numerical simulations. In particular, we would like to point out that this does not only hold for the overall score, but also for the individual entries of the confusion matrix. This confirms the excellent controllability and robustness of the experimental system. Specifically, hand gestures (first three rows) are recognized perfectly. Fast spatial movements of the subjects—jogging and running—are more challenging to differentiate because, for example, one subject's running may be very similar to another subject's jogging. Therefore, the confusion matrices reflect several errors between these two classes. The walking action is also similar in appearance, but slower on the temporal scale, hence it is more accurately classified by the system.

Discussion and conclusion

In this work, we present a photonic video-processing system for human-action recognition. Unlike the recent advances in computer

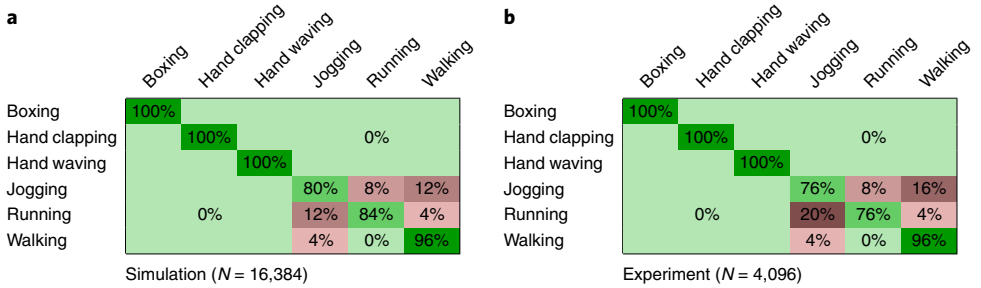

Fig. 5 | Confusion matrices with the best performance. a,**b**, Confusion matrices for the best scores obtained numerically (**a**) and experimentally (**b**).

vision, relying on deep learning, we have implemented a shallow recurrent neural network—a reservoir computer—which not only simplifies the training process, but allows one to realize the network in hardware, such as photonic systems, inherently leveraging the parallelism of optics. We demonstrate a highly flexible optical experiment that allows us to accommodate a very large number of physical nodes (N = 16,384), with the potential of scaling up to hundreds of thousands of nodes, thus offering considerable advantages in terms of parallelism and speed, and for realization of the crucial vector-matrix products. The natural scalability of the proposed photonic architecture could be further exploited to process multiple video feeds in parallel by allocated various regions of the SLM screen to independent reservoir computers, each processing a specific video stream with the strategy described in this Article.

Finally and despite the simplicity of the system, its performance on the KTH dataset is comparable to state-of-the-art deep approaches and superior to gradient-optimized LSTM networks. Our optical information processing system is particularly well suited for data that is already in the optical domain, such as image and video processing, as studied here. This work thus proposes a hardware solution to video information processing that could potentially outperform deep learning in terms of training time and complexity.

Methods

Basic principles of RC. A typical discrete-time reservoir computer was discussed in section 'Photonic reservoir computer', equation (1). The dynamics of the reservoir is determined by the matrices W_{ij}^{res} and B_{ij}, both time-independent and drawn from a random distribution with zero mean. The reservoir computer produces M output signals $y_i(n)$, corresponding to the M output nodes (in this work, M = 6), given by a linear combination of the states of its internal variables

$$y_l(n) = \sum_{j=0}^{N-1} W_{lj}^{\mathrm{out}} x_j(n) \tag{2}$$

where W_{lj}^{out} are the readout weights, trained either offline (using standard linear regression methods, such as the ridge-regression algorithm[34] used here) or online[29], to minimize the NMSE between the output signal $y(n)$ and the target signal $d(n)$, given by

$$\mathrm{NMSE} = \frac{\left\langle (y(n) - d(n))^2 \right\rangle}{\left\langle (d(n) - \langle d(n) \rangle)^2 \right\rangle} \tag{3}$$

Physical modelling of the photonic reservoir computer. The state variable $x_i(n)$ of the ith photonic neuron at discrete time step n is the 10 bit quantified optical intensity W_{ij}^{res} detected by the camera. We use the structure of the set-up to determine the evolution of this state variable. It starts with a linear transformation by the network adjacency matrix and the addition of a masked input data. This relation is used to update the 8 bit quantified phase value vector loaded in the SLM's controller according to the following equation:

$$\left\lfloor \phi_i(n+1) \right\rfloor_8 = \left\lfloor \sum_{j=0}^{N-1} W_{ij}^{\mathrm{res}} x_j(n) + \sum_{j=0}^{K-1} B_{ij} u_j(n) \right\rfloor_8 \tag{4}$$

with W_{ij}^{res} and B_{ij} being the reservoir adjacency matrix and input mask, respectively. The phase of the ith SLM's macro-pixel is nonlinearly converted into an intensity value because of the peculiar polarization configuration of the optical arm comprising the liquid crystal on silicon (LCoS) SLM and two polarizers rotated by 45° with respect to the orientation of the SLM's liquid crystals in their resting state. Using the theoretical framework of Jones calculus (see ref. [35] for more details), we can easily show that $f_{\mathrm{NL},I}(\cdot) = \left\lfloor I_0 \sin^2\left(\left\lfloor \cdot \right\rfloor_8\right) \right\rfloor_{10}$. Hence, the evolution equation for the ith neuron's state reads

$$x_i(n+1) = f_{\mathrm{NL},I}\left(\sum_{j=0}^{N-1} W_{ij}^{\mathrm{res}} x_j(n) + \sum_{j=0}^{K-1} B_{ij} u_j(n) \right) \tag{5}$$

with $f_{\mathrm{NL},I}(\cdot) = \left\lfloor I_0 \sin^2\left(\left\lfloor \cdot \right\rfloor_8\right) \right\rfloor_{10}$ being the nonlinear function and I_0 the uniform optical intensity illuminating (and reflected from) the SLM and camera. Without loss of generality, I_0 can be normalized at a unitary value. Here, a reservoir output is defined by

$$y_l(n) = \sum_{j=0}^{N-1} W_{lj}^{\mathrm{out}} x_j(n) \tag{6}$$

with W_{out} the readout matrix of trainable coefficients for the six outputs of the reservoir (one output per action to recognize).

An alternative approach is to consider the 8 bit quantified, macro-pixel phase shift $\left\lfloor \phi_i(n) \right\rfloor_8$ induced by the SLM's liquid crystals as the state variable $x_i(n)$ of the ith neuron at discrete time n. In this modelling scenario, the dynamics of the system can also read

$$x_i(n+1) = \left\lfloor \sum_{j=0}^{N-1} W_{ij}^{\mathrm{res}} f_{\mathrm{NL},\phi}(x_j(n)) + \sum_{j=0}^{K-1} B_{ij} u_j(n) \right\rfloor_8 \tag{7}$$

with W_{ij}^{res} being the nonlinear function and I_0 the uniform optical intensity illuminating (and reflected from) the SLM and camera. Without loss of generality, I_0 can be normalized at a unitary value. In this case, the reservoir output is defined by

$$y_l(n) = \sum_{j=0}^{N-1} W_{lj}^{\mathrm{out}} f_{NL,\phi}\left(x_j(n)\right) \tag{8}$$

Hyperparameters. The dynamics of the reservoir can be optimized for a given task by tuning several control parameters. The input mask B_{ij} is drawn from a random distribution over the interval [−1, 1] and then multiplied by a coefficient β, called the input gain, which controls the amplitude of the external input signal, that is, the degree of perturbation of the reservoir. The generation of the interconnection matrix W_{ij}^{res} requires two additional parameters: a scaling factor γ and a density ρ. Because the echo-state network paradigm[36] requires the interconnection matrix to be sparse, W_{ij}^{res} is generated from a random distribution over the interval [−1, 1] with $\rho \times N^2$ non-zero elements. The matrix is then multiplied by a global scaling factor γ, which determines the strength of connections between different neurons within the network. The diagonal elements of W_{ij}^{res}, which define the feedback of each neuron to itself, are defined separately. Given that we want all neurons to exhibit the same internal dynamics, we set the diagonal elements of W_{ij}^{res} to α, a parameter called the feedback gain.

In summary, the dynamics of the system are defined by four hyperparameters—the input gain β, the feedback gain α, the interconnection gain γ and the interconnection density ρ. The optimization of hyperparameters is performed through grid search (that is, a parameter sweep)—an exhaustive search through all possible combinations of manually specified values of all the parameters. Table 2 presents the intervals used for the optimization, and the optimal values for selected reservoir sizes, considered both numerically and experimentally.

Hyperparameter optimizations have shown the input and feedback gains to be important variables; that is, accurate values are required to obtain the best performance, while the characteristics of the interconnection matrix play a minor role. We managed to obtain comparable scores with significantly different W_{ij}^{res} matrices in terms of density and amplitude of the off-diagonal elements.

The KTH dataset. The original KTH video database[23] contains four different scenarios. In this work, for simplicity, we limited the dataset to the first scenario,

Table 2 | Optimal hyperparameters for reservoirs of different sizes

Parameter	Symbol	Search values	Optimal for $N=1{,}024$		Optimal for $N=16{,}384$	
			Num.	Exp.	Num.	Exp.
Feedback gain	α	0.1–1.5	0.8	0.8	0.6	0.3
Input gain	β	0.0001–1	0.01	0.1	0.16	0.16
Interconnectivity gain	γ	0.0001–1	0.1	0.1	0.001	0.001
Interconnectivity density	ρ	0.0001–1	0.01	0.001	0.001	0.001

referred to as 's1', containing outdoor videos (illustrated in Fig. 1a). All videos were recorded over a homogeneous background with a static camera and 25 f.p.s., then downsampled to a spatial resolution of 160×120 pixels. Each single action movie had a length of 4 s, on average. The subjects repeated each action four times. In total, our dataset contained 25×6×4=600 sequences for each combination of 25 subjects, 6 actions and 4 repetitions. The DivX-compressed videos were first uncompressed and split into 160×120 greyscale frames. Different sequences varied in length and contained between 24 and 239 frames.

Histograms of oriented gradients. The HOG algorithm, introduced by Dalal and Triggs[24], is based on scale-invariant features transform (SIFT) descriptors[37]. To calculate a HOG descriptor, horizontal and vertical gradients are first computed by filtering the image with the following kernels[25]:

$$G_x = (-1, 0, 1) \quad \text{and} \quad G_y = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix} \tag{9}$$

Then, magnitude $m(x, y)$ and orientation $\theta(x, y)$ of gradients are computed for each pixel, using

$$m(x,y) = \sqrt{D_x^2 + D_y^2} \quad \text{and} \quad \theta(x,y) = \arctan\left(\frac{D_y}{D_x}\right) \tag{10}$$

where D_x and D_y are approximations of the horizontal and vertical gradients, respectively.

The creation of histograms starts with the division of the image into small cells. Each cell is assigned a histogram of typically nine bins, corresponding to angles 0, 20, 40, …160, and containing the sums of magnitudes of the gradients within the cell. The main purpose of this operation is to provide a compact, yet truthful description of a patch of an image. That is, a typical cell of 8×8 greyscale pixels is described with nine numbers instead of 64. As gradients of an image are sensitive to overall lighting, the algorithm is completed with block normalization, by dividing the histograms by their euclidean norm computed over bigger-sized blocks.

The computation of HOG features was performed in Matlab, using the built-in `extractHOGFeatures` function, individually for each frame of every sequence, with a cell size of 8×8 and a block size of 2×2. Given the frame size of 160×120 pixels, the function returns 19×14×4×9=9,576 features per frame. Figure 1b illustrates the resulting gradients superimposed on top of a video-frame from the KTH dataset.

Data availability

The KTH dataset can be downloaded from http://www.nada.kth.se/cvap/actions/. The numerical and experimental data can be downloaded from the data folder in our GitHub repository: https://github.com/pantonik/rc_slm_kth/ (https://doi.org/10.5281/zenodo.3474559).

Code availability

The code used in this study can be downloaded from the `scripts` folder in our GitHub repository: https://github.com/pantonik/rc_slm_kth (https://doi.org/10.5281/zenodo.3474559).

Received: 28 April 2019; Accepted: 7 October 2019;
Published online: 12 November 2019

References

1. Wu, D., Sharma, N. & Blumenstein, M. Recent advances in video-based human action recognition using deep learning: a review. In *2017 International Joint Conference on Neural Networks (IJCNN)* https://doi.org/10.1109/ijcnn.2017.7966210 (IEEE, 2017).
2. Moeslund, T. B. & Granum, E. A survey of computer vision-based human motion capture. *Comput. Vis. Image Underst.* **81**, 231–268 (2001).
3. Moeslund, T. B. in *Virtual Interaction: Interaction in Virtual Inhabited 3D Worlds* (eds Qvortrup, L. et al.) 221–234 (Springer, 2001).
4. Vrigkas, M., Nikou, C. & Kakadiaris, I. A. A review of human activity recognition methods. *Front. Robot. AI* **2**, 28 (2015).
5. Jaeger, H. Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. *Science* **304**, 78–80 (2004).
6. Maass, W., Natschläger, T. & Markram, H. Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Comput.* **14**, 2531–2560 (2002).
7. Lukoševičius, M. & Jaeger, H. Reservoir computing approaches to recurrent neural network training. *Comput. Sci. Rev.* **3**, 127–149 (2009).
8. Appeltant, L. et al. Information processing using a single dynamical node as complex system. *Nat. Commun.* **2**, 468 (2011).
9. Paquot, Y. et al. Optoelectronic reservoir computing. *Sci. Rep.* **2**, 287 (2012).
10. Larger, L. et al. Photonic information processing beyond turing: an optoelectronic implementation of reservoir computing. *Opt. Express* **20**, 3241 (2012).
11. Martinenghi, R., Rybalko, S., Jacquot, M., Chembo, Y. K. & Larger, L. Photonic nonlinear transient computing with multiple-delay wavelength dynamics. *Phys. Rev. Lett.* **108**, 244101 (2012).
12. Larger, L. et al. High-speed photonic reservoir computing using a time-delay-based architecture: million words per second classification. *Phys. Rev. X* **7**, 011015 (2017).
13. Duport, F., Schneider, B., Smerieri, A., Haelterman, M. & Massar, S. All-optical reservoir computing. *Opt. Express* **20**, 22783 (2012).
14. Brunner, D., Soriano, M. C., Mirasso, C. R. & Fischer, I. Parallel photonic information processing at gigabyte per second data rates using transient states. *Nat. Commun.* **4**, 1364 (2013).
15. Vinckier, Q. et al. High-performance photonic reservoir computer based on a coherently driven passive cavity. *Optica* **2**, 438 (2015).
16. Akrout, A. et al. Parallel photonic reservoir computing using frequency multiplexing of neurons. Preprint at https://arxiv.org/abs/1612.08606 (2016).
17. Vandoorne, K. et al. Experimental demonstration of reservoir computing on a silicon photonics chip. *Nat. Commun.* **5**, 3541 (2014).
18. Triefenbach, F., Jalalvand, A., Schrauwen, B. & Martens, J.-P. Phoneme recognition with large hierarchical reservoirs. In *Advances in Neural Information Processing Systems Proceedings* 2307–2315 (NIPS, 2010).
19. The 2006/07 Forecasting Competition for Neural Networks and Computational Intelligence http://www.neural-forecasting-competition.com/NN3/ (2006).
20. Antonik, P., Haelterman, M. & Massar, S. Brain-inspired photonic signal processor for generating periodic patterns and emulating chaotic systems. *Phys. Rev. Appl.* **7**, 054014 (2017).
21. Bueno, J. et al. Reinforcement learning in a large-scale photonic recurrent neural network. *Optica* **5**, 756 (2018).
22. Hagerstrom, A. M. et al. Experimental observation of chimeras in coupled-map lattices. *Nat. Phys.* **8**, 658–661 (2012).
23. Schuldt, C., Laptev, I. & Caputo, B. Recognizing human actions: a local SVM approach. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004* https://doi.org/10.1109/icpr.2004.1334462 (IEEE, 2004).
24. Dalal, N. & Triggs, B. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)* https://doi.org/10.1109/cvpr.2005.177 (IEEE, 2005).
25. Bahi, H. E., Mahani, Z., Zatni, A. & Saoud, S. A robust system for printed and handwritten character recognition of images obtained by camera phone. In *WSEAS Transactions on Signal Processing* (WSEAS, 2015).
26. Pearson, K. L. III On lines and planes of closest fit to systems of points in space. *Lond. Edinb. Dubl. Phil. Mag. J. Sci.* **2**, 559–572 (1901).
27. Hotelling, H. Analysis of a complex of statistical variables into principal components. *J. Educ. Psychol.* **24**, 417–441 (1933).
28. Smith, L. I. *A Tutorial on Principal Components Analysis*. Technical report, Univ. Otago (2002).
29. Antonik, P. et al. Online training of an opto-electronic reservoir computer applied to real-time channel equalization. *IEEE Trans. Neural Netw. Learn. Systems* **28**, 2686–2698 (2017).
30. Psaltis, D. & Farhat, N. Optical information processing based on an associative-memory model of neural nets with thresholding and feedback. *Opt. Lett.* **10**, 98 (1985).
31. Jhuang, H. *A Biologically Inspired System for Action Recognition*. PhD thesis, Massachusetts Institute of Technology (2007).
32. Grushin, A., Monner, D. D., Reggia, J. A. & Mishra, A. Robust human action recognition via long short-term memory. In *The 2013 International Joint Conference on Neural Networks (IJCNN)* https://doi.org/10.1109/ijcnn.2013.6706797 (IEEE, 2013).
33. Gilbert, A., Illingworth, J. & Bowden, R. Action recognition using mined hierarchical compound features. *IEEE Trans. Pattern Anal. Mach. Intell.* **33**, 883–897 (2011).
34. Tikhonov, A. N, Goncharsky, A, Stepanov, V. & Yagola, A. G. *Numerical Methods for the Solution of Ill-posed Problems* (Springer, 1995).
35. Saleh, B. E. A. & Teich, M. C. *Fundamental of Photonics* 3rd edn (Wiley, 2019).
36. Jaeger, H. The 'echo state' approach to analysing and training recurrent neural networks—with an Erratum note. *GMD Report* **148**, 1–47 (2001).

37. Lowe, D. G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **60**, 91–110 (2004).
38. Yadav, G. K., Shukla, P. & Sethfi, A. Action recognition using interest points capturing differential motion information. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* https://doi.org/10.1109/icassp.2016.7472003 (IEEE, 2016).
39. Shi, Y., Zeng, W., Huang, T. & Wang, Y. Learning deep trajectory descriptor for action recognition in videos using deep neural networks. In *2015 IEEE International Conference on Multimedia and Expo (ICME)* https://doi.org/10.1109/icme.2015.7177461 (IEEE, 2015).
40. Kovashka, A. & Grauman, K. Learning a hierarchy of discriminative space–time neighborhood features for human action recognition. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* https://doi.org/10.1109/cvpr.2010.5539881 (IEEE, 2010).
41. Baccouche, M., Mamalet, F., Wolf, C., Garcia, C. & Baskurt, A. in *Sequential Deep Learning for Human Action Recognition* 29–39 (Springer, 2011).
42. Ali, K. H. & Wang, T. Learning features for action recognition and identity with deep belief networks. In *2014 International Conference on Audio, Language and Image Processing* https://doi.org/10.1109/icalip.2014.7009771 (IEEE, 2014).
43. Wang, H., Klaser, A., Schmid, C. & Liu, C.-L. Action recognition by dense trajectories. In *2011 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)* https://doi.org/10.1109/cvpr.2011.5995407 (IEEE, 2011).
44. Liu, J. & Shah, M. Learning human actions via information maximization. In *2008 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* https://doi.org/10.1109/cvpr.2008.4587723 (IEEE, 2008).
45. Sun, X., Chen, M. & Hauptmann, A. Action recognition via local descriptors and holistic features. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops* https://doi.org/10.1109/cvprw.2009.5204255 (IEEE, 2009).
46. Veeriah, V., Zhuang, N. & Qi, G.-J. Differential recurrent neural networks for action recognition. In *2015 IEEE International Conference on Computer Vision (ICCV)* https://doi.org/10.1109/iccv.2015.460 (IEEE, 2015).
47. Shu, N., Tang, Q. & Liu, H. A bio-inspired approach modeling spiking neural networks of visual cortex for human action recognition. In *2014 International Joint Conference on Neural Networks (IJCNN)* https://doi.org/10.1109/ijcnn.2014.6889832 (IEEE, 2014).
48. Laptev, I., Marszalek, M., Schmid, C. & Rozenfeld, B. Learning realistic human actions from movies. In *2008 IEEE Conference on Computer Vision and Pattern Recognition* https://doi.org/10.1109/cvpr.2008.4587756 (IEEE, 2008).
49. Klaeser, A., Marszalek, M. & Schmid, C. A spatio-temporal descriptor based on 3D-gradients. In *Proceedings of the British Machine Vision Conference 2008* https://doi.org/10.5244/c.22.99 (British Machine Vision Association, 2008).
50. Ji, S., Xu, W., Yang, M. & Yu, K. 3D convolutional neural networks for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**, 221–231 (2013).
51. Escobar, M.-J. & Kornprobst, P. Action recognition via bio-inspired features: the richness of center–surround interaction. *Comput. Vis. Image Underst.* **116**, 593–605 (2012).

Acknowledgements

The authors thank the creators of the KTH dataset for making the videos publicly available. This work was supported by AFOSR (grants nos. FA-9550-15-1-0279 and FA-9550-17-1-0072), Région Grand-Est and the Volkswagen Foundation via the NeuroQNet Project.

Author contributions

D.B., N.M. and D.R. designed and managed the study. P.A., N.M. and D.R. realized the experimental set-up. P.A. performed the numerical simulations and the experimental campaigns. P.A., N.M. and D.R. prepared the manuscript. All authors discussed the results and reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to P.A. or D.R.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.