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

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

[1]LMOPS EA 4423 Laboratory, CentraleSupélec & Université de Lorraine, F-57070 Metz, France
[2]FEMTO-ST Institute/Optics Department, CNRS & University Bourgogne Franche-Comté, F-25030 Besançon, France
[*]Corresponding authors: piotr.antonik@centralesupelec.fr, damien.rontani@centralesupelec.fr

ABSTRACT

The recognition of human actions in video streams is a challenging task in computer vision, with cardinal applications in e.g. brain-computer interface and surveillance. Deep learning has shown remarkable results recently, but can be found hard to use in practice, as its training requires large datasets and special purpose, energy-consuming hardware. In this work, we propose a scalable photonic neuro-inspired architecture based on the reservoir computing paradigm, capable of recognising video-based human actions with state-of-the-art accuracy. Our experimental optical setup comprises off-the-shelf components, and implements a large parallel recurrent neural network that is easy to train and can be scaled up to hundreds of thousands of nodes. This work paves the way towards simply reconfigurable and energy-efficient photonic information processing systems for real-time video processing.

1 Introduction

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 of this research field are the potential applications, which can be found in various 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 concerning system-control make use of the captured motions to provide control functionality in games, virtual environments, or to control remote devices[3]. The detailed automatic analysis of motions could be used in clinical studies of e.g. orthopedic patients, or to help athletes improve their performance[2].

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

In this work, we propose an optical signal processing system for classification of video-based human actions. The idea of optical computing has been investigated for decades as photons propagate without generating heat or signal degradation due to induction and capacitive effects, and thus promise a high level of parallelism in e.g. optical communications. Neural networks would heavily benefit from parallel signal transmission, which, as shown by the increasing usage of optical interconnects in modern computing systems, is one of the strong suits of photonics. An optical approach could thus allow one 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 paradigm*. Reservoir Computing (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 optimised by solving a system of linear equations[7]. Furthermore, as less 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 reservoir computing in electronics[8], optoelectronics[9–12], optics[13–16], and integrated on chip[17] is comparable to other digital algorithms on a series of benchmark tasks, such as wireless channel equalisation[5], phoneme recognition[18] and prediction of future evolution of financial time series[19]. Finally, it was shown that the readout layer of photonic reservoir computers can be implemented optically and trained using a digital micro-mirror device[20].

In this paper, we present an optoelectronic reservoir computer, inspired by Refs.[20,21]. The system is based on the phase modulation of a spatially extended planar wave by means of a spatial light modulator (SLM). Our scheme offers a notable parallelisation 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 setup can accommodate a reservoir of 16,384 nodes, while the physical limitation of the concept is set to as high as $262,144$ neurons. The input and the output layers, as well as the recurrence of the network, are realised digitally in this work.

The system is benchmarked on the popular KTH database[22], which contains video recordings of 6 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 uniform background. At the pre-processing stage, the histograms of oriented gradients (HOG) algorithm[23] (described later) is used to extract spatial and shape information from individual video frames. The photonic reservoir computer is used to classify the 6 motions given the resulting HOG features.

The setup is evaluated both experimentally and in simulations. The numerical model was design 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 optimised 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 $90.7\% - 95.6\%$ 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, that benefits from the intrinsic parallelism of photonics, and thus offers a highly-scalable and, potentially, energy-efficient neural network.

2 Results

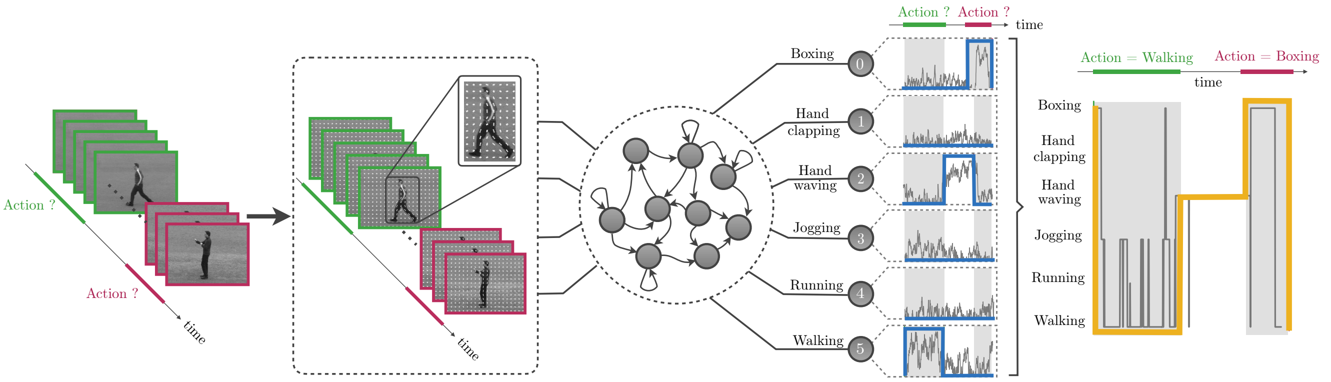
Before presenting the results of this study, we introduce the video-based human action classification task in the context of reservoir computing, and then present the experimental setup. The theory of reservoir computing can be found in the Methods section.

2.1 Classification of human action with a reservoir computer

The principles of the human action recognition task in the context of reservoir computing are illustrated in Fig. 1. In this work, we used the popular KTH database of human actions[22], publicly available online, which consists of video recordings of 6 different motions (walking, jogging, running, boxing, hand waving, and hand clapping) performed by 25 subjects. Each subjects performs each motion 4 times, which results in a dataset of 600 video sequences of variables lengths, ranging from 24 to 239 frames. More details on the video properties of the dataset can be found in the Methods section. All videos are concatenated together and split into individual frames, giving the raw video stream (see Fig. 1(a)), carried forward to the pre-processing stage (Fig. 1(b)).

Feature extraction is a common approach in computer vision to provide 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 state of the art. Therefore, we turned to the histograms of oriented gradients algorithm, introduced by Dalal and Triggs[23]. There, 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 localisation and detection of objects (see e.g.[24]). 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 local object appearance and shape can often be expressed well enough by distribution of local intensity gradients or edges' directions. The HOG algorithm is further discussed in the Methods section. To reduce the number of resulting HOG features and simplify computations, we apply the principal component analysis (PCA)[25,26] based on the covariance method[27] . We choose to keep the first 2,000 components (out of 9,576), who's eigenvalues account for 91.6% of the total variability in the data.

The training of the reservoir computer, illustrated in Fig. 1(c), was performed frame-wise on a subset of 450 video sequences, each one containing a single motion sequence; 150 sequences were used to evaluate the performance of the system. Figure 1(d) illustrates the 6 binary classifiers, introduced to distinguish the motions: 6 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. 1(e), 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.



(a) video stream (b) HOG preprocessing (c) photonic computer (d) binary outputs (e) winner-takes-all classification

Figure 1. Scheme of principle of how our reservoir computer solves the human action classification task. The video input stream (a) is a concatenation of the 600 video sequences available in the KTH dataset; 450 sequences were used for training, and 150 for testing. The input stream undergoes a pre-processing stage (b), where the HOG algorithm is applied to each individual frame. The dimensionality reduction through the PCA is not illustrated in this figure. The selected features are fed into the photonic reservoir computer (c), trained to classify each individual frame. This is achieved by defining 6 binary output nodes (d), one for each action class, that are trained to output 1 for a frame of the corresponding class and 0 for the others. Target outputs are shown in blue. The frame-wise classification (e) is obtained by selecting the node with the maximum output, *i.e.* 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, that 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-gray left-hand side region). However, since more frames in the sequence are classified as walking (74.5%) than jogging (23.6%), the entire sequence is correctly classified as walking.

During the training, the NMSE cost function (see Eq. 3) was used to minimise the error between the reservoir output and the target class. In this study, it was noted that the final classification did not require the output of the correct class to be as close as possible to "1", while the others being close to "0". Since 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, lower NMSE does not necessarily mean less classification errors. Therefore, we used a different error metric based on the confusion matrix[22]. 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 classified into the 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 6 actions have been recognised with a 100% accuracy.

2.2 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 $n \in \mathbb{Z}$, as given by

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

where $f_{\mathrm{NL},I}$ is a nonlinear function (in this work, $f_{\mathrm{NL},I}(x) = \lfloor \sin^2(\lfloor x \rfloor_8) \rfloor_{10}$), W_{ij}^{res} is a $N \times N$ matrix of interconnecting weights between the neurons of the neural network, $u_j(n)$ is an input with K dimensions, and B_{ij} is a $N \times K$ matrix of input weights, often referred to as the *input mask*. Further information on the principles of reservoir computing and the properties of B_{ij} and W_{ij}^{res} can be found in the Methods section.

Our experimental setup implements Eq. 1 and is schematised in Fig. 2. 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 Eq. 1. It is powered by a green LED source at 532 nm (Thorlabs M530L3) set to a power level of 10.5 mW. The choice of the wavelength was based on the availability of

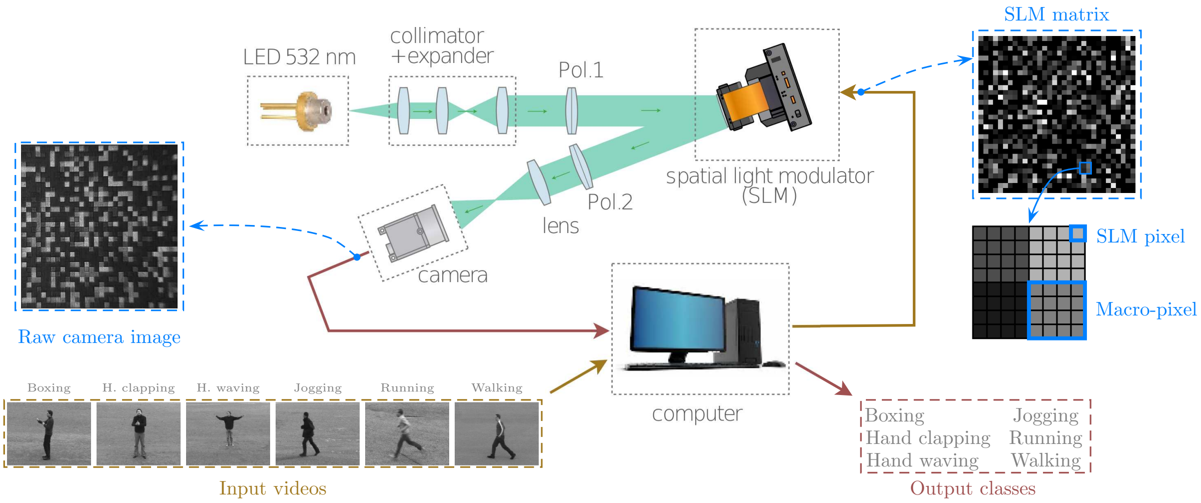


Figure 2. Illustration of the experimental setup, composed of an optical arm, connected to a computer. The output of a green LED (LED 532 nm) is collimated and expanded (collimator+expander), then polarised (Pol. 1), and used to illuminate the surface of the spatial light modulator (SLM). The latter is being imaged by a high-speed camera (camera) through a second polariser (Pol. 2) and an imaging lens (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, i.e. the SLM matrix. Groups of small individual pixels of the SLM are combined into larger macro-pixels, that 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.

optical components and the ease of use and calibration of the setup in the visible spectrum. The optical power was adjusted so as to provide sufficient illumination of the SLM to generate the highest contrast, with adequate exposure settings on the camera. The output beam is linearly polarised, collimated and expanded to roughly 17 mm in diameter to evenly illuminate the entire 7.68 mm × 7.68 mm surface of the spatial light modulator (Meadowlark XY Phase P512 – 0532 with 8-bit resolution). In simplified terms, a SLM is a variable, spatially resolved wave plate: its index of refraction along the slow axis can be decreased electronically. That is, a linearly polarised illumination beam, parallel to the slow axis of the SLM, is reflected with a phase-only modulation. If a beam is parallel to the fast axis instead, one would observe no modulation with the variable voltage. In this setup, an illumination beam oriented at 45° with respect to the slow axis provides equal optical field components to the fast and the slow axis of the SLM. After reflection, the former remains unchanged, while the latter undergoes a phase modulation. A second polariser transforms the phase difference between the two components into intensity modulations, which are in turn imaged onto a high-speed camera (Allied Vision Mako U-130B with 10-bit resolution). The imaging-system is optimised for a compromise between imaging resolution and the field of view's extend.

The computer runs 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 W_{ij}^{res} are generated randomly at the beginning of the experiment. At each discrete timestep n, the input to the nonlinear function $\sum W_{ij}^{res} x_j(n) + \sum B_{ij} u_j(n)$ is computed, and the resulting matrix is loaded onto the SLM device. The camera then records a picture of the SLM through the imaging lens and the polarising optics. The raw image is 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 Eq. 1. The states are rearranged into a vector x_j and used to compute the next SLM matrix at timestep $n+1$.

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

The speed of the setup 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, the system is capable of processing 2 video frames per second. In the case of a small reservoir ($N = 1,024$), the matrix multiplication $\sum W_{ij}^{res} x_j(n)$ (see Eq. 1) requires less computations, and the processing speed is increased up to 7 frames per second. The use of Matlab at this stage is a deliberate choice, as it lends considerable flexibility to the setup, for example testing different pre-processing techniques, reservoir topologies, and output decision-making layers by simply changing the code, i.e. without reconfiguration of the optical setup. The system's speed limitation can 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 e.g.[28]). More importantly, matrix can equally be offloaded to fully parallel optics[20,29]. 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 fps.

2.3 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 3 shows the resulting performance for different reservoir sizes in terms of the classification score (introduced in Sec. 2.1), computed during the testing phase. The hyperparameters were optimised from scratch for each reservoir size, and independently in simulations and experiments. Details on the optimisation approach can be found in Sec. 4.3. In numerical simulations, we investigated the performance with different random input and interconnection weights. We performed 5 distinctive simulations for each reservoir size with different random wights and full optimisation of the hyperparameters. Blue bars display the average performance and the error bars show the standard deviations, i.e. 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 runtimes are due to the optimisation of the hyperparameters through grid search (see Sec. 4.3). With one set of hyperparameters, the experiment processes the full dataset (i.e. both the training and test stages) in 1.6 to 5.5 hours, 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, that 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 (the details can be found in the respective 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, thus directly comparable to our results. In terms of performance, the photonic RC is short by 4.7% from the best results on the *s1* scenario[42], but outperforms the SVM approach in terms of processing speed by a factor of ten. The training time of our system is significantly shorter than deep approaches[44], and comparable to the SVM method with hierarchical compound features[33]. Our photonic approach thus offers a performing, more flexible, and easy to train classification system. Furthermore, the recent development of integrated photonic reservoir computers[17] could give raise to very energy-efficient optical processors.

Table 2 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 confusion-matrix's individual entries. This confirms the excellent controllability and robustness of the experimental system. Specifically, hand gestures (first three rows) are perfectly recognised. Fast spatial movements of the subjects – jogging and running – are more challenging to differentiate because, for instance, 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.

3 Discussion & conclusion

In this work, we present a photonic video-processing system for human-action recognition. Unlike the recent advances in computer 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 realise the network in hardware, such as photonic systems, inherently leveraging the parallelism of optics. We demonstrate a highly flexible optical experiment that allows 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 realisation of the crucial vector-matrix products. The natural scalability of the proposed photonic architecture could be further exploited to process multiple video feeds in

					Performance	
Authors	Method	Database split	Training time	Processing speed	*s1* scenario	Full database
Yadav et al.[30]	IP + SVM	80%-20%	–	–	–	98.20%
Shi et al.[31]	DTD, DNN	9-16	–	–	–	95.6%
Kovashka et al.[32]	BoW + SVM	8-8-9	–	–	–	94.53%
Gilbert et al.[33]	HCF + SVM	LOOCV	~ 5.6 h	24 fps	–	94.5%
Baccouche et al.[34]	CNN & RNN	16-9	–	–	–	94.39%
Ali and Wang[35]	DBN & SVM	50%-20%-30%	–	–	–	94.3%
Wang et al.[36]	DT + SVM	16-9	–	–	–	94.2%
Liu et al.[37]	MMI + SVM	LOOCV	–	–	–	94.15%
Sun et al.[38]	FT + SVM	auto	–	–	–	94.0%
Veeriah et al.[39]	Differential RNN	16-9	–	–	–	93.96%
Shu et al.[40]	SNN	9-16	–	–	95.3%	92.3%
Laptev et al.[41]	FT + SVM	8-8-9	–	–	–	91.8%
Jhuang[42]	StC_2 + SVM	16-9	–	0.4 fps	96.0%	91.6%
Klaeser et al.[43]	3D Grad + SVM	8-8-9	–	–	–	91.4%
This work	**Photonic RC**	**75%-25%**	1.6 – 5.5 h	$2-7$ **fps**	**91.3%**	–
Grushin et al.[44]	LSTM	16-9	1 day	$12-15$ fps	–	90.7%
Ji et al.[45]	3DCNN	8-8-9	–	–	–	90.02%
Escobar et al.[46]	MT cells	16-9	–	–	74.63%	–
Schuldt et al.[22]	FT + SVM	8-8-9	–	–	–	71.83%

Table 1. Performance of various state-of-the-art digital approaches compared to our best experimental result. 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, e.g. 16 subjects for training, 9 for the test) or three groups (training, validation and test, e.g. 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, considered in this work.

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 paper.

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-optimised LSTM networks. Our optical information processing system is particularly well suited for the data that is already in the optical domain, such as image and video processing, studied here. This work thus proposes a hardware solution to video information processing, that could outperform deep learning in terms of training time and complexity.

4 Methods

4.1 Basic principles of reservoir computing

A typical discrete-time reservoir computer was discussed in Sec. 2.2, Eq. 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

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

where W_{lj}^{out} are the readout weights, trained either offline (using standard linear regression methods, such as the ridge-regression algorithm[47] used here), or online[28], in order to minimise the normalised mean square error (NMSE) between the output signal $y(n)$ and the target signal $d(n)$, given by

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

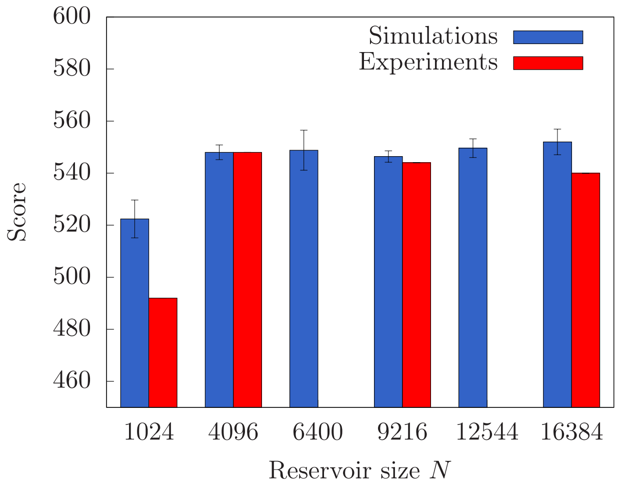


Figure 3. Performance of our photonic neuro-inspired architecture on the human action classification task. Different reservoir sizes have been investigated numerically (blue) and experimentally (red). The error bars on the numerical results show the score variability (standard deviation) with 5 different input masks. Experimental variability could not be measured because of the long runtime of the experiment.

	Boxing	Hand clapping	Hand waving	Jogging	Running	Walking
Boxing	100%					
Hand clapping		100%			0%	
Hand waving			100%			
Jogging				80%	8%	12%
Running		0%		12%	84%	4%
Walking				4%	0%	96%

Simulation ($N = 16,384$)

	Boxing	Hand clapping	Hand waving	Jogging	Running	Walking
Boxing	100%					
Hand clapping		100%			0%	
Hand waving			100%			
Jogging				76%	8%	16%
Running		0%		20%	76%	4%
Walking				4%	0%	96%

Experiment ($N = 4,096$)

Table 2. Confusion matrices with the best performance.

4.2 Physical modeling of the photonic reservoir computer

The state variable $x_i(n)$ of the i-th photonic neuron at discrete time step n is the 10-bit quantified optical intensity $\lfloor I_i(n) \rfloor_{10}$ detected by the camera. We use the structure of the setup 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

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

with W_{ij}^{res} and B_{ij} the reservoir adjacency matrix and input mask, respectively. The phase of the i-th SLM's macro-pixel is nonlinearly converted into an intensity value because of the peculiar polarisation configuration of the optical arm comprising the LCoS SLM and two polarisers rotated by 45 degrees with respect to the orientation of the SLM's liquid crystals in their resting state. Using the theoretical framework of Jones calculus (see Ref.[48] for more details), we can easily show that $\lfloor I_i(n+1) \rfloor_{10} = \lfloor I_0 \sin^2(\lfloor \phi_i(n+1) \rfloor_8) \rfloor_{10}$. Hence, the evolution equation for the i-th neuron's state reads

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

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

Table 3. Optimal hyperparameters for reservoirs of different sizes.

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

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

with W_{out} the readout matrix of trainable coefficients for the 6 outputs of the reservoir (one output per action to recognise).

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

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

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

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

4.3 Hyperparameters

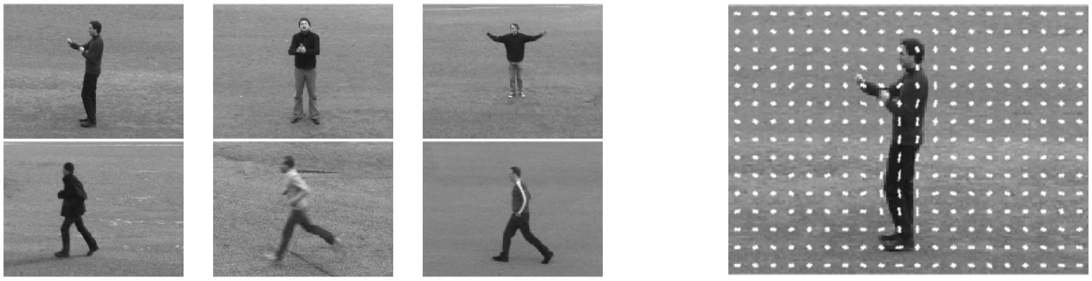
The dynamics of the reservoir can be optimised for a given task by tuning several control parameters. The input mask B_{ij} is drawn from a random distribution over the interval $[-1, 1]$ and then multiplied by a coefficient β, called the *input gain*, which controls the amplitude of the external input 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. Since the echo-state network paradigm[49] 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. Since 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 optimisation of hyperparameters is performed through grid search (*i.e.* parameter sweep) – an exhaustive search through all possible combinations of manually specified values of all the parameters. Table 3 presents the intervals used for the optimisation, and the optimal values for selected reservoir sizes, considered both numerically and experimentally.

Hyperparameters optimisation have shown the input and feedback gains to be important variables, i.e. accurate values are required to obtain the best performance, while the characteristics of 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.

4.4 The KTH dataset

The original KTH video database[22] contains four different scenarios. In this work, for simplicity, we limited the dataset to the first scenario, referred to as "s1", containing outdoor videos (illustrated in Fig. 4a). All videos were recorded over homogeneous



(a) Examples of KTH frames

(b) Example of HOG features

Figure 4. **(a)** Examples of action frames from the KTH database, from left to right: boxing, hand clapping, hand waving, jogging, running, and walking. Six different subjects are illustrated out of the total of 25. All videos have been 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 visualised 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 to capture the pose of the subject.

background with a static camera and 25 fps, then downsampled to the spatial resolution of 160 × 120 pixels. Each single action movie has a length of four seconds in average. The subjects repeat each action 4 times. In total, our dataset contains $25 \times 6 \times 4 = 600$ sequences for each combination of 25 subjects, 6 actions, and 4 repetitions. The DIVX-compressed videos are first uncompressed and split into 160 × 120 grayscale frames. Different sequences vary in length and contain between 24 and 239 frames.

4.5 Histograms of oriented gradients

The histograms of oriented gradients (HOG) algorithm, introduced by Dalal and Triggs[23], is based on Scale-Invariant Features transform (SIFT) descriptors[50]. To calculate a HOG descriptor, first, horizontal and vertical gradients are computed by filtering the image with the following kernels[24]:

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

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

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

where D_x and D_y are the approximations of 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 9 bins, corresponding to angles $0, 20, 40, \ldots 160$, 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 grayscale pixels is described with 9 numbers instead of 64. As gradients of an image are sensitive to overall lighting, the algorithm is completed with block normalisation, 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 $19 \times 14 \times 4 \times 9 = 9576$ features per frame. Figure 4b illustrates the resulting gradients superimposed on top of a video-frame from the KTH dataset.

Data availability statement

The KTH dataset can be downloaded here: `http://www.nada.kth.se/cvap/actions/`. The numerical and experimental data can be downloaded here: `addlink`.

Code availability statement

The code used in this study can be downloaded here: `addlink`.

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)*, DOI: 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, DOI: 10.1006/cviu.2000.0897 (2001).
3. Moeslund, T. B. Interacting with a virtual world through motion capture. In *Virtual Interaction: Interaction in Virtual Inhabited 3D Worlds*, 221–234, DOI: 10.1007/978-1-4471-3698-9_11 (Springer London, 2001).
4. Vrigkas, M., Nikou, C. & Kakadiaris, I. A. A review of human activity recognition methods. *Front. Robotics AI* **2**, DOI: 10.3389/frobt.2015.00028 (2015).
5. Jaeger, H. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science* **304**, 78–80, DOI: 10.1126/science.1091277 (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, DOI: 10.1162/089976602760407955 (2002).
7. Lukoševičius, M. & Jaeger, H. Reservoir computing approaches to recurrent neural network training. *Comput. Sci. Rev.* **3**, 127–149, DOI: 10.1016/j.cosrev.2009.03.005 (2009).
8. Appeltant, L. *et al.* Information processing using a single dynamical node as complex system. *Nat. Commun.* **2**, DOI: 10.1038/ncomms1476 (2011).
9. Paquot, Y. *et al.* Optoelectronic reservoir computing. *Sci. Reports* **2**, DOI: 10.1038/srep00287 (2012).
10. Larger, L. *et al.* Photonic information processing beyond turing: an optoelectronic implementation of reservoir computing. *Opt. Express* **20**, 3241, DOI: 10.1364/oe.20.003241 (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**, DOI: 10.1103/physrevlett.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**, DOI: 10.1103/physrevx.7.011015 (2017).
13. Duport, F., Schneider, B., Smerieri, A., Haelterman, M. & Massar, S. All-optical reservoir computing. *Opt. Express* **20**, 22783, DOI: 10.1364/oe.20.022783 (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**, DOI: 10.1038/ncomms2368 (2013).
15. Vinckier, Q. *et al.* High-performance photonic reservoir computer based on a coherently driven passive cavity. *Optica* **2**, 438, DOI: 10.1364/optica.2.000438 (2015).
16. Akrout, A. *et al.* Parallel photonic reservoir computing using frequency multiplexing of neurons. *arXiv:1612.08606* (2016).
17. Vandoorne, K. *et al.* Experimental demonstration of reservoir computing on a silicon photonics chip. *Nat. Commun.* **5**, DOI: 10.1038/ncomms4541 (2014).
18. Triefenbach, F., Jalalvand, A., Schrauwen, B. & Martens, J.-P. Phoneme recognition with large hierarchical reservoirs. In *Advances in neural information processing systems*, 2307–2315 (2010).
19. The 2006/07 forecasting competition for neural networks & computational intelligence. http://www.neural-forecasting-competition.com/NN3/ (2006).
20. Bueno, J. *et al.* Reinforcement learning in a large-scale photonic recurrent neural network. *Optica* **5**, 756, DOI: 10.1364/optica.5.000756 (2018).
21. Hagerstrom, A. M. *et al.* Experimental observation of chimeras in coupled-map lattices. *Nat. Phys.* **8**, 658–661, DOI: 10.1038/nphys2372 (2012).
22. Schuldt, C., Laptev, I. & Caputo, B. Recognizing human actions: a local SVM approach. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, DOI: 10.1109/icpr.2004.1334462 (IEEE, 2004).

23. Dalal, N. & Triggs, B. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, DOI: 10.1109/cvpr.2005.177 (IEEE, 2005).

24. Bahi, H. E., Mahani, Z., Zatni, A. & Saoud, S. A robust system for printed and handwritten character recognition of images obtained by camera phone. Tech. Rep. (2015).

25. Pearson, K. LIII. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, Dublin Philos. Mag. J. Sci.* **2**, 559–572, DOI: 10.1080/14786440109462720 (1901).

26. Hotelling, H. Analysis of a complex of statistical variables into principal components. *J. Educ. Psychol.* **24**, 417–441, DOI: 10.1037/h0071325 (1933).

27. Smith, L. I. A tutorial on principal components analysis. Tech. Rep. (2002).

28. Antonik, P. *et al.* Online training of an opto-electronic reservoir computer applied to real-time channel equalization. *IEEE Transactions on Neural Networks Learn. Syst.* **28**, 2686–2698, DOI: 10.1109/tnnls.2016.2598655 (2017).

29. Psaltis, D. & Farhat, N. Optical information processing based on an associative-memory model of neural nets with thresholding and feedback. *Opt. Lett.* **10**, 98, DOI: 10.1364/ol.10.000098 (1985).

30. 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)*, DOI: 10.1109/icassp.2016.7472003 (IEEE, 2016).

31. 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)*, DOI: 10.1109/icme.2015.7177461 (IEEE, 2015).

32. 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*, DOI: 10.1109/cvpr.2010.5539881 (IEEE, 2010).

33. Gilbert, A., Illingworth, J. & Bowden, R. Action recognition using mined hierarchical compound features. *IEEE Transactions on Pattern Analysis Mach. Intell.* **33**, 883–897, DOI: 10.1109/tpami.2010.144 (2011).

34. Baccouche, M., Mamalet, F., Wolf, C., Garcia, C. & Baskurt, A. Sequential deep learning for human action recognition. In *Lecture Notes in Computer Science*, 29–39, DOI: 10.1007/978-3-642-25446-8_4 (Springer Berlin Heidelberg, 2011).

35. 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*, DOI: 10.1109/icalip.2014.7009771 (IEEE, 2014).

36. Wang, H., Klaser, A., Schmid, C. & Liu, C.-L. Action recognition by dense trajectories. In *CVPR 2011*, DOI: 10.1109/cvpr.2011.5995407 (IEEE, 2011).

37. Liu, J. & Shah, M. Learning human actions via information maximization. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, DOI: 10.1109/cvpr.2008.4587723 (IEEE, 2008).

38. 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*, DOI: 10.1109/cvprw.2009.5204255 (IEEE, 2009).

39. Veeriah, V., Zhuang, N. & Qi, G.-J. Differential recurrent neural networks for action recognition. In *2015 IEEE International Conference on Computer Vision (ICCV)*, DOI: 10.1109/iccv.2015.460 (IEEE, 2015).

40. 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)*, DOI: 10.1109/ijcnn.2014.6889832 (IEEE, 2014).

41. Laptev, I., Marszalek, M., Schmid, C. & Rozenfeld, B. Learning realistic human actions from movies. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, DOI: 10.1109/cvpr.2008.4587756 (IEEE, 2008).

42. Jhuang, H. *A biologically inspired system for action recognition*. Ph.D. thesis, Massachusetts Institute of Technology (2007).

43. Klaeser, A., Marszalek, M. & Schmid, C. A spatio-temporal descriptor based on 3d-gradients. In *Procedings of the British Machine Vision Conference 2008*, DOI: 10.5244/c.22.99 (British Machine Vision Association, 2008).

44. 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)*, DOI: 10.1109/ijcnn.2013.6706797 (IEEE, 2013).

45. Ji, S., Xu, W., Yang, M. & Yu, K. 3d convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis Mach. Intell.* **35**, 221–231, DOI: 10.1109/tpami.2012.59 (2013).

46. Escobar, M.-J. & Kornprobst, P. Action recognition via bio-inspired features: The richness of center–surround interaction. *Comput. Vis. Image Underst.* **116**, 593–605, DOI: 10.1016/j.cviu.2012.01.002 (2012).

47. Tikhonov, A. N., Goncharsky, A., Stepanov, V. & Yagola, A. G. *Numerical methods for the solution of ill-posed problems*, vol. 328 (Springer Netherlands, 1995).

48. Saleh, B. E. A. & Teich, M. C. *Fundamental of Photonics* (Wiley, 2019), 3rd edn.

49. Jaeger, H. The "echo state" approach to analysing and training recurrent neural networks - with an Erratum note. *GMD Rep.* **148** (2001).

50. Lowe, D. G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **60**, 91–110, DOI: 10.1023/b:visi.0000029664.99615.94 (2004).

Acknowledgements

The authors thank the creators of the KTH dataset for making the videos publicly available. This work was supported by AFOSR (grants No. FA-9550-15-1-0279 and FA-9550-17-1-0072), Région Grand-Est, and the Volkswagen Foundation via the NeuroQNet Project.

5 Author contributions statement

D.B, N.M., and D.R designed and managed the study. P.A., N.M., and D.R. realised the experimental setup. 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.

6 Additional information

Competing interests The authors declare no competing interests.