Frontiers in NEUROMORPHIC ENGINEERING

Research Article
4 December 2015

Benchmarking Spike-Based Visual Recognition: a Dataset and Evaluation

Qian Liu [1,*], Garibaldi Pineda-García [1], Evangelos Stromatias [1], Teresa Serrano-Gotarredona [2], and Steve Furber [1]

[1] *Advanced Processor Technologies Research Group, School of Computer Science, University of Manchester, Manchester, United Kingdom*
[2] *Instituto de Microelectrónica de Sevilla (IMSE- CNM-CSIC), Sevilla, Spain*

Correspondence*:
Qian Liu
SpiNNaker, Advanced Processor Technologies Research Group, School of Computer Science, The University of Manchester, Oxford Road, Manchester, M13 9PL, United Kingdom, qianl.liu-3@manchester.ac.uk

ABSTRACT

To gain a better understanding of the brain and build biologically-inspired computers, increasing attention is being paid to research into spike-based neural computation. Within the field, the visual pathway and its hierarchical organisation have been extensively studied within the primate brain. Spiking Neural Networks (SNNs) inspired by the understanding of observed biological structure and function have been successfully applied to visual recognition/classification tasks. In addition, implementations on neuromorhpic hardware have made large-scale networks run in (or even faster than) real time, and accessible on mobile robots. Neuromorphic sensors, e.g. silicon retinas, are able to feed such a mobile system with real-time visual stimuli. A new series of vision benchmarks for spike-based neural processing are now needed to quantitatively measure progress within this rapidly advancing field. We propose that a large dataset of spike-based visual stimuli is needed to provide a baseline for comparisons on SNN models and algorithms, and some benchmarking network models are also required to validate the accuracy and cost of these neuromorphic hardware platforms.

First of all, an initial NE (Neuromorphic Engineering) dataset of input stimuli based on standard computer vision benchmarks consisting of digits (from the MNIST database) is presented according to the current research on spike-based image recognition. Within this dataset, all images are centre aligned and having similar scale. We describe how we intend to expand this dataset to fulfil the needs of upcoming research problems. For instance, the data should provide cases to measure position-, scale-, and viewing-angle invariance. The data are in Address-Event Representation (AER) format which is widely used in the neuromorphic engineering field unlike conventional images. These spike trains are produced by various techniques: rate-based Poisson spike generation, rank order encoding and recorded output from a silicon retina with both flashing and oscillating input stimuli. Furthermore a complementary evaluation methodology is also presented to assess both model-level and hardware-level performance. Finally, we provide two SNN models to validate their classification capabilities and to assess the performances of their hardware implementations as tentative benchmarks.

With this dataset we hope to (1) promote meaningful comparison between algorithms in the field of neural computation, (2) allow comparison with conventional image recognition methods,

(3) provide an assessment of the state of the art in spike-based visual recognition, and (4) help researchers identify future directions and advance the field.

Keywords: Benchmarking, Vision Dataset, Evaluation, Neuromorphic Engineering, Spiking Neural Networks

1 INTRODUCTION

With rapid developments in neural engineering, researchers are approaching the aims of understanding brain functions and building brain-like machines using this knowledge (Furber and Temple, 2007). As a fast growing field, neuromorphic engineering has provided biologically-inspired sensors such as DVS (Dynamic Vision Sensor) silicon retinas (Serrano-Gotarredona and Linares-Barranco, 2013; Delbruck, 2008; Yang et al., 2015; Posch et al., 2014), which are good examples of low-cost visual processing thanks to their event-driven and redundancy-reducing style of computation. Moreover, SNN simulation tools (Davison et al., 2008; Gewaltig and Diesmann, 2007; Goodman and Brette, 2008) and neuromorphic hardware platforms (Furber et al., 2014; Schemmel et al., 2010; Merolla et al., 2014) have been developed to allow exploration of the brain by mimicking its functions and developing large-scale practical applications (Eliasmith et al., 2012). Particularly for visual processing, the central visual system consists of several cortical areas which are placed in a hierarchical pattern according to anatomical experiments (Felleman and Van Essen, 1991). Fast object recognition takes place in the feed-forward hierarchy of the ventral pathway, one of the two central visual pathways, which mainly handles the "What" tasks. Experiments have revealed that the information is unfolded along the ventral stream to the IT (Inferior Temporal) cortex (DiCarlo et al., 2012). Inspired by the explicit biological study of the central visual pathway, SNNs models have successfully been adapted to computer vision tasks.

Riesenhuber and Poggio (1999) proposed a quantitative modelling framework of object recognition with position-, scale- and view-invariance based on the units of MAX-like operations. The cortical-like model has been analysed on several datasets (Serre et al., 2007). And recently Fu et al. (2012) reported that their SNN implementation of the framework was capable of facial expression recognition with a classification accuracy (CA) of 97.35% on the JAFFE dataset (Lyons et al., 1998) which contains 213 images of 7 facial expressions posed by 10 individuals. They employed simple integrate-and-fire neurons with rank order coding (ROC) where the earliest pre-synaptic spikes have the strongest impact on the post synaptic potentials. According to Van Rullen and Thorpe (2002), the first wave of spikes carry explicit information through the ventral stream and in each stage meaningful information is extracted and spikes are regenerated. Using one spike per neuron, Delorme and Thorpe (2001) reported 100% and 97.5% accuracies on the face identification task over changing contrast and luminance training (40 individuals × 8 images) and testing data (40 individuals × 2 images) respectively.

The Convolutional Neural Network (CNN), also known as the *ConvNet* developed by LeCun et al. (1998), is a well applied model of such a cortex-like framework. An early Convolutional Spiking Neural Network (CSNN) model identified faces of 35 persons with a CA of 98.3% exploiting simple integrate and fire neurons (Matsugu et al., 2002). Another CSNN model (Zhao et al., 2015) was trained and tested both with DVS raw data and Leaky Integrate-and-Fire (LIF) neurons. It was capable of recognising three moving postures with a CA of about 99.48% and 88.14% on the MNIST-DVS dataset (see Chapter 2.2). As one step forward, Camunas-Mesa et al. (2012) implemented a convolution processor module in hardware which could be combined with a DVS for high-speed recognition tasks. The inputs of the ConvNet were continuous spike events instead of static images or frame-based videos. The chip detected four suits of a 52 card deck while the cards were fast browsed in only 410 ms. Similarly, a real-time gesture recognition model (Liu and Furber, 2015) was implemented on a neuromorphic system with a DVS as a front-end and a SpiNNaker (Furber et al., 2014) machine as the back-end where LIF neurons built up the ConvNet configured with biological parameters. In this study's largest configuration, a network of 74,210 neurons and 15,216,512 synapses used 290 SpiNNaker cores in parallel and reached 93.0% accuracy.

Deep Neural Networks (DNNs) together with deep learning are the most exciting research fields in vision recognition. The spiking deep network has great potential to combine remarkable performance with the energy efficient training and running. In the initial stage of the research, the study was focused on converting off-line trained deep network to SNNs (O'Connor et al., 2013). The same network initially implemented on a FPGA achieved a CA of 92.0% (Neil and Liu, 2014), while a later implementation on SpiNNaker scored 95.0% (Stromatias et al., 2015a). Recent attempts have contributed to better translation by utilising modified units in a ConvNet (Cao et al., 2015) and tuning the weights and thresholds (Diehl et al., 2015)). The later paper claims a state-of-the-art performance (99.1% on the MNIST dataset) comparing to original ConvNet. The current trend of training Spiking DNNs on-line using biologically-plausible learning methods is also promising. An event driven Contrastive Divergence (CD) training algorithm for RBMs (Restricted Boltzmann Machines) was proposed for Deep Belief Networks (DBN) using LIF neurons with STDP (Spike-Timing-Dependent Plasticity) synapses and verified on MNIST (91.9%) (Neftci et al., 2013).

STDP as a biological learning process is applied to vision tasks. Bichler et al. (2012) demonstrated an unsupervised STDP learning model to classify car trajectories captured with a DVS retina. A similar model was tested on a Poissonian spike presentation of the MNIST dataset achieving a performance of 95.0% (Diehl and Cook, 2015). Theoretical analysis (Nessler et al., 2013) showed that unsupervised STDP was able to approximate a stochastic version of Expectation Maximization, a powerful learning algorithm in machine learning. The computer simulation achieved 93.3% CA on MNIST and could be implemented in a memrisitve device (Bill and Legenstein, 2014).

Despite the promising research on SNN-based vision recognition, there is no commonly used database in the format of spikes. In the studies listed above, all the vision data used are in one of the following formats: (1) the grey-scale raw values of images; (2) rate-based spike trains according to pixel intensities created by various Poissonian generators; (3) unpublished DVS recorded spike-based videos. However in the field of conventional non-spiking computer vision, there are a few datasets playing important roles at different times and with various objectives. The MNIST (LeCun et al., 1998) dataset is a subset of the NIST hand written digits dataset. Due to its straightforward target of classifying real-world images, the plain format of binary data and the simple patterns, MNIST has been one of the most popular datasets in computer vision for over 20 years. ImageNet (Deng et al., 2009) was put forward to provide researchers with a large-scale image database, which currently contains 14,197,122 images. The dataset is a well-recognised benchmark test for the deep learning community, and many attempts have been made to improve the performance of machine learning algorithms on this dataset, for example (Krizhevsky et al., 2012). As a good example of a database catching up with state-of-the-art technologies, Microsoft COCO aims to solve three problems: objects categorisation, context understanding and spatial labelling, in scene understanding by providing large-scale datasets (300,000+ images). Similar examples could be found in video datasets. Two early benchmarks, the KTH (Schüldt et al., 2004) and Weizmann (Blank et al., 2005) datasets, have been used extensively in the past decade. These videos were produced with scripted behaviours in a controlled environment ("in the lab"). The YouTube Action Dataset (Liu et al., 2009) targets recognising realistic actions from videos "in the wild". The main challenge relies on the massive variations due to the moving camera, background clutter, viewing angles, illuminations and so on.

As a consequence, a new series of spike-based vision datasets is now needed to quantitatively measure progress within this rapidly advancing field and to provide fair competition resources for researchers. Apart from using spikes instead of the frame-based data of conventional computer vision, there are new concerns of evaluating neuromorphic vision in tasks other than recognition accuracy. Therefore a common metric of performance evaluation on spike-based vision is also required to specify the measurements of algorithms and models. Different assessments should be taken into consideration when implementing models on neuromorphic hardware, especially the trade-offs between simulation time, precision and power consumption. Thus benchmarking neuromorphic hardware with various network models will reveal the advantages and disadvantages of different platforms. In this paper we propose a large dataset of spike-based visual stimuli, NE, and its complementary evaluation methodology. The dataset expands and evolves as research develops and new problems are introduced.

Section 2 defines the purpose and protocols of the proposed dataset, describes the sub-datasets and their generation methods, and demonstrates the evaluation methodology in accordance with the dataset.

Moreover, two SNN models are provided as examples of benchmarking hardware platforms in Section 3. Section 4 summarises the paper and discusses future work.

2 MATERIALS AND METHODS

2.1 GUIDING PRINCIPLES

The NE database we propose here is a developing and evolving dataset consisting of various spike-based representations of images and videos. The spikes are either generated from spike encoding methods which covert images or frames of videos into spike trains, or recorded from DVS silicon retinas. The spike trains are in the format of AER (Mahowald, 1992) data, which could easily be used in both event-driven computer simulations and neuromorphic systems. The AER is originally proposed as a communication protocol using time-multiplexing to replace the connection wires for each connected neuron pair. The AER data contains two columns, and each row consists of the time stamp of a spike and the address of the neuron which generated the spike. With the NE dataset we hope:

- *to promote meaningful comparisons of algorithms in the field of spiking neural computation.* The NE dataset provides a unified format of AER data to meet the demands of spike-based visual stimuli. It also encourages researchers to publish and contribute their data to build up the NE dataset. The training and testing sets have to be disjoint and also of similar quality and quantity.
- *to allow comparison with conventional image recognition methods.* It asks the dataset to support this comparison with spiking versions of existing vision datasets. Thus, conversion methods are required to transform datasets of images and frame-based videos to spike stimuli. With growing knowledge of biological vision, new methodologies and algorithms are welcomed to present these conventional datasets with spikes in more biological ways.
- *to provide an assessment of the state of the art in spike-based visual recognition on neuromorphic hardware.* In order to reveal the advantages of neuromorphic engineering, not only a spike based dataset but also an appropriate evaluation methodology is needed. In accordance with the idea of an evolving dataset, the evaluation methodology develops accordingly as a constantly perfected process.
- *to help researchers identify future directions and advance the field.* The development of the dataset and its evaluation will introduce new challenges to the neuromorphic engineering community. However, an easily solved problem turns out to be a tuning competition, while a far more difficult problem is not appropriate to bring meaningful assessment. So suitable problems should be added continuously to promote future research.

2.2 THE DATASET: NE15-MNIST

The name of the first proposed dataset in the benchmarking system is NE15-MNIST which stands for Neuromorphic Engineering 2015 on MNIST. The original MNIST dataset is downloaded from the website[1] of THE MNIST DATABASE of handwritten digits (LeCun et al., 1998). The NE15-MNIST is converted into a spiking version of the original dataset consisting of four subsets which were generated for different purposes:

- *Poissonian* to benchmarking existing methods of rate-based spiking models.
- *FoCal (Filter overlap Correction)* to promote the study of spatio-temporal algorithms applied to recognition tasks using few input spikes.

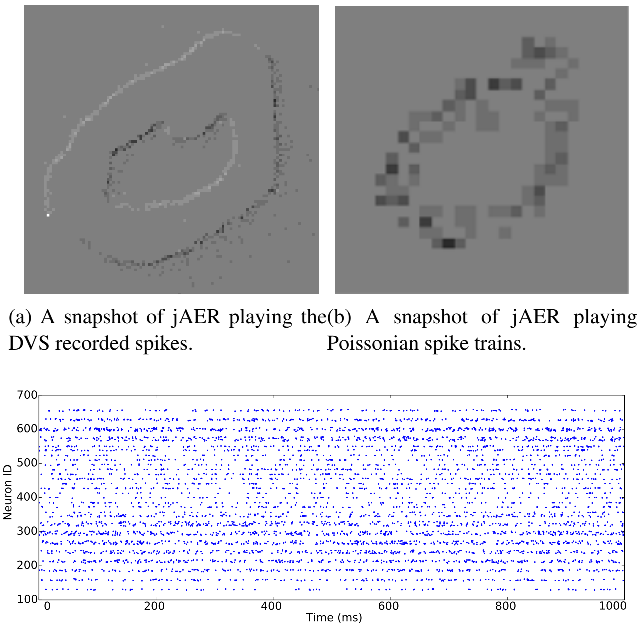
[1] http://yann.lecun.com/exdb/mnist/

- *DVS recorded flashing input* to encourage research on fast recognition methods which are found in the primate visual pathway.
- *DVS recorded moving input* to trigger the study of algorithms targeting on continuous input from real-world sensors and to implement them on mobile neuromorphic robots.

The dataset can be found in the GitHub repository at: https://github.com/qian-liu/benchmarking.

2.3 DATA DESCRIPTION

Two file formats are supported in the dataset: jAER format (Delbruck, 2008) (.dat or .aedat), and binary file in NumPy .npy format. The AER interface has been widely used in neuromorphic systems, especially for vision sensors. The spikes are encoded as time events with corresponding addresses to convey information. The spikes in jAER format, both recorded from a DVS retina and artificially generated, can be displayed in jAER software. Figure 1a is a snapshot of the software displaying an .aedat file which is recorded by a DVS retina (Serrano-Gotarredona and Linares-Barranco, 2013). The resolution of the DVS recorded data is 128×128. The other format of spikes used is a list of spike source arrays in PyNN (Davison et al., 2008), a description language for building spiking neuronal network models. Python code for converting one file format to and from the other is also provided. The duration of artificially generated data could be configured using the Python code provided, while the recorded data aiming at different tasks vary in duration: 1 s for the flashing input, and 2.5 s for the moving input.



(a) A snapshot of jAER playing the DVS recorded spikes.

(b) A snapshot of jAER playing Poissonian spike trains.

(c) The raster plot of the Poissonian spike trains.

Figure 1: Snapshots of jAER software playing spike presented videos. The same image of digit "0" is transformed to spikes by DVS recording and the Poissonian generation respectively. A raster plot of the Poissonian spike trains is also provided.

2.3.1 Poissonian In the cortex, the timing of spikes is highly irregular (Squire and Kosslyn, 1998). It can be interpreted that the inter-spike interval reflects a random process driven by the instantaneous firing rate. If the generation of each spike is assumed to be independent of all the other spikes, the spike train is seen as a Poisson process. The spiking rate can be estimated by averaging the pooled responses of the neurons.

As stated above, rate coding is exclusively used in presenting images with spikes. The spiking rate of each neuron is in accordance with its corresponding pixel intensity. Instead of providing exact spike arrays, we share the Python code for generating the spikes. Every recognition system may require different spiking rates and various lengths of their durations. The generated Poissonian spikes can be in the formats of both jAER and PyNN spike source array. Thus, it is easy to visualise the digits and also to build spiking neural networks. Because different simulators generate random Poissonian spike trains with various mechanisms, languages and codes, using the same dataset enables performance evaluation on different simulators without the interference created by non-unified input. The same digit displayed in Figure 1(a) is converted to Poissonian spike trains, see Figure 1(b). The raster plot can be found in Figure 1(c), indicating the intensities of the pixels.

2.3.2 Rank-Order-Encoding A different way of encoding spikes is using a rank-order code; this means keeping just the order in which those spikes were fired and disregarding the exact timing. Rank-ordered spike trains have been used in vision tasks under a biological plausibility constraint, making them a viable way of image encoding for neural applications (Van Rullen and Thorpe, 2001; Sen and Furber, 2009; Masmoudi et al., 2010).

Rank-ordered encoding can be performed using an algorithm known as the FoCal algorithm (Sen and Furber, 2009). It models the foveal pit region, the highest resolution area of the retina, with four ganglion cell layers that show a centre-surround behaviour (Kolb, 2003). In order to simulate these layers, four discrete 2D convolutions are performed. The centre-surround behaviour of the ganglion cells is modelled using Differences of Gaussians (DoG).

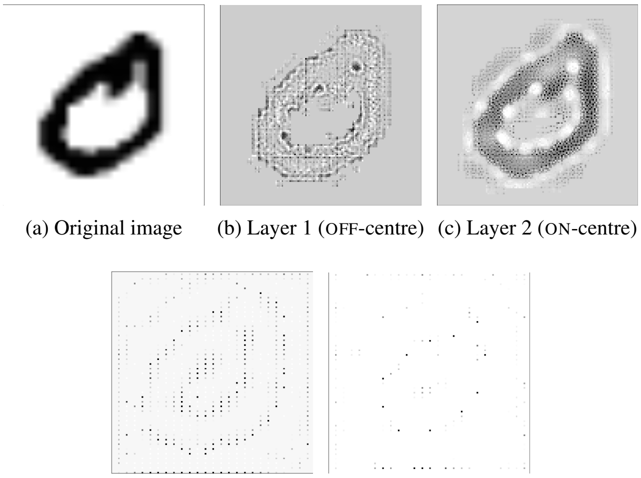
$$DoG_w(x,y) = \pm\frac{1}{2\pi\sigma_{w,c}^2} e^{\frac{-(x^2+y^2)}{2\sigma_{w,c}^2}} \mp \frac{1}{2\pi\sigma_{w,s}^2} e^{\frac{-(x^2+y^2)}{2\sigma_{w,s}^2}} \tag{1}$$

where $\sigma_{w,c}$ and $\sigma_{w,s}$ are the standard deviation for the centre and surround components of the DoG at layer w. The signs will be $(-,+)$ if the ganglion cell has an OFF-centre behaviour and $(+,-)$ if it has an ON-centre one. Table 1 describes the parameters used to compute the convolution kernels at each scale w.

Table 1. Simulation parameters for ganglion cells

Layer	Centre type	Matrix width	Centre std. dev. (σ_c)	Surround std. dev. (σ_s)	Sampling resolution (cols,rows)
1	OFF	3	0.8	$6.7 \times \sigma_c$	1, 1
2	ON	11	1.04	$6.7 \times \sigma_c$	1, 1
3	OFF	61	8	$4.8 \times \sigma_c$	5, 3
4	ON	243	10.4	$4.8 \times \sigma_c$	5, 3

Every pixel value in the convolved images (Figure 2) is inversely proportional to a spike emission time with respect to the presentation of the image (i.e. the higher the pixel value, the sooner the spike will be sent out.)


(a) Original image (b) Layer 1 (OFF-centre) (c) Layer 2 (ON-centre)

(d) Layer 3 (OFF-centre) (e) Layer 4 (ON-centre)

Figure 2: Results of correcting the spikes from the simulated ganglion cell layers using the FoCal algorithms.

Since DoGs where used as a means to encode the image, and they are not an orthogonal basis, the algorithm also performs a redundancy correction step, it does so by adjusting the convolved image's pixel value according to the correlation between convolution kernels (Alg. 1).

Algorithm 1 FoCal, redundancy correction

```
procedure CORRECTION(coeffs $C$, correlations $Q$)
    $N \leftarrow \emptyset$                                  ▷ Corrected coefficients
    repeat
        $m \leftarrow max(C)$                                 ▷ Obtain maximum from $C$
        $M \leftarrow M \cup m$                               ▷ Add maximum to $M$
        $C \leftarrow C \setminus m$                          ▷ Remove maximum from $C$
        for all $c \in C$ do                                  ▷ Adjust all remaining $c$
            if $Q(m,c) \neq 0$ then                           ▷ Adjust only near
                $c \leftarrow c - m \times Q(m,c)$
            end if
        end for
    until $C = \emptyset$
    return $M$
end procedure
```

After the correction step, the most important information can be recovered using only the first 30% of the spikes (Sen and Furber, 2009). These significant spikes are shown in Figure 3, assuming that each spike will be generated 1 ms apart. Neurons in Layer 1 emit spikes faster and in larger quantities than any other layer, making it the most important one. Layers 2 and 3 have few spikes, this is due to the large convolution kernels used to simulate the ganglion cells. One of the main advantages of ROC is that neurons will only spike once, this can be seen particularly well in these two layers. Layers 0 and 1 encode fine details, while layers 2 and 3 result in blob like features.

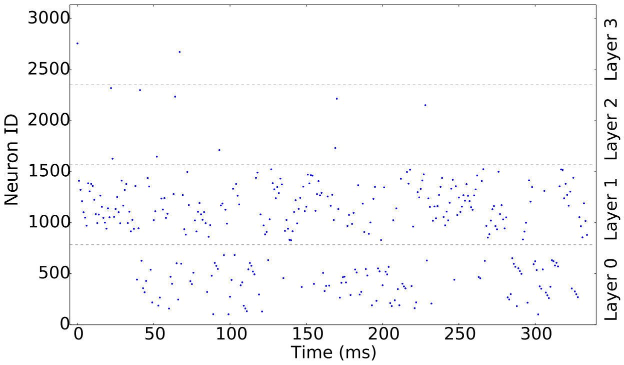


Figure 3: First 30% of the rank-order encoded spikes produced with FoCal.

Figure 4 shows the reconstruction results for the two stages of the algorithm. On Figure 4(b) the reconstruction was applied after the convolution but without the FoCal correction, a blurry image is the result of redundancy in the spike representation. A better reconstruction can be obtained after Algorithm 1 has been applied, the result is shown in Figure 4(c).

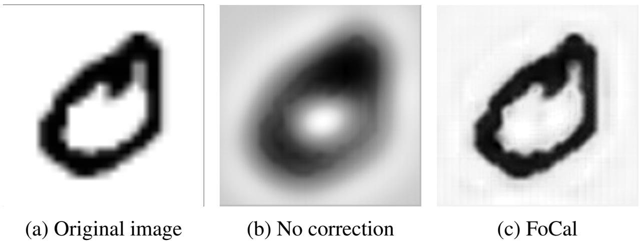

(a) Original image (b) No correction (c) FoCal

Figure 4: Reconstruction result comparison.

The source Python scripts to transform images to ROC spike trains, and to convert the results into AER and PyNN's spike source array can be found in the dataset's website.

2.3.3 DVS Sensor Output with Flashing Input The purpose of including the subset of DVS recorded flashing digits is to promote the application of Rank-Order-Coding to DVS output, and accelerate the fast on-set recognition by using just the beginning part of spike trains within less than 30 ms.

Each digit and a blank image was shown alternately and each display lasted one second. The digits were displayed on an LCD monitor in front of the DVS retina (Serrano-Gotarredona and Linares-Barranco, 2013) and were placed in the centre of the visual field of the camera. Since there are two polarities of the spikes: 'ON' indicates the increase of the intensity while 'OFF' reflects the opposite, there are 'ON' and 'OFF' flashing recordings respectively per digit. In Figure 5, the burstiness of the spikes is illustrated where most of the spikes occur in a 30 ms slot. In total, the subset of the database contains $2 \times 60,000$ recordings for training and $2 \times 10,000$ for testing.

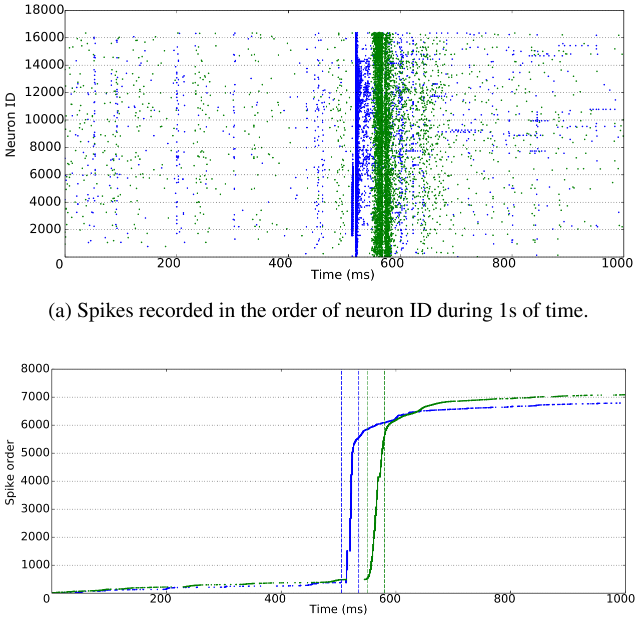
2.3.4 DVS Sensor Output with Moving Input In order to address the problems of position- and scale-invariance, a subset of DVS recorded moving digits is presented.

MNIST digits were scaled to three different sizes, by using smooth interpolation algorithms to increase their size from the original 28x28 pixel size, and displayed on the monitor with slow motion. The same DVS (Serrano-Gotarredona and Linares-Barranco, 2013) used in Section 2.3.3 captured the movements of the digits and generated spike trains for each pixel of its 128×128 resolution. A total of 30,000 recordings were made: 10 digits, at 3 different scales, 1000 different handwritten samples for each.

2.4 PERFORMANCE EVALUATION

A complementary evaluation methodology is essential to provide common metrics and assess both the model-level and hardware-level performance.

2.4.1 Hardware-Independent Evaluation First of all it is desirable for researchers to specify whether they add any preprocessing either to images or spikes. Filtering the raw input may ease the classification/recognition task while adding noise may require stronger robustness of the model. Secondly, as with the evaluation on conventional artificial neural networks, a description of the network characteristics is most welcome since it is the basis for the overall performance. Furthermore, sharing the designs may inspire fellow scientists to bring new points of view to the problem and generate a positive feedback loop where everybody wins. The network description should include the topology, and the neural and synaptic models. The network topology defines the number of neurons used for each layer, and the connections between layers and neurons. Some researchers make use of extra non-neural classifiers, sometimes to aid the design, others to enhance the output of the network. Any particulars on this subject are greatly appreciated. It is essential to state the type of neural and synaptic model (e.g. current-based LIF neuron) exploited in the network and the parameters configuring them, because neural activities differ greatly between various configurations. Thirdly, the learning procedure determines the recognition capability of a network model. A clear distinction has always been made between supervised,



(a) Spikes recorded in the order of neuron ID during 1s of time.

(b) Spikes plotted in the sequence of appearing time during 1s of time. Bursty spikes apeer in slots less than 30 ms.

Figure 5: The bursty of spikes is illustrated where most of the spikes occur in a 30 ms slot. Blue for ’ON’ events and green for ’OFF’.

Table 2. Hardware independent comparison

	Preprocessing	Network	Training	Recognition
Brader et al. (2007)	None	Two layer, LIF neurons	Semi-supervised, STDP, calcium LTP/LTD	96.5%
Beyeler et al. (2013)	None	V1 (edge), V4 (orientation), and competitive decision, Izhikevich neurons	Semi-supervised, STDP, calcium LTP/LTD	91.6% 300 ms per test
Neftci et al. (2013)	Thresholding	Two layer RBM, LIF neurons	Event-driven contrastive divergence, supervised	91.9% 1 s per test
Diehl and Cook (2015)	None	Two layers, LIF neurons, inhibitory feedback	Unsupervised, exp. STDP, $3,000,000$ s of training $200,000$ s per iteration	95%
Diehl et al. (2015)	None	ConvNet or Fully connected, LIF neurons	Off-line trained with ReLU, weight normalization	99.1% (ConvNet), 98.6% (Fully connected); 0.5 s per test
Zhao et al. (2015)	Thresholding or DVS	Simple (Gabor), Complex (MAX) and Tempotron	Tempotron, supervised	Thresholding 91.3%, 11 s per test DVS 88.1%, 2 s per test
This paper	None	Four layer RBM, LIF neurons	Off-line trained, unsupervised	94.94% 16 ms latency
This paper	None	Fully connected decision layer, LIF neurons	K-means clusters, Supervised STDP $18,000$ s of training	92.98% 1 s per test 10.70 ms latency

semi-supervised and unsupervised learning. A detailed description of new proposed spike-based learning rules will be a great contribution to the field due to the lack of spatio-temporal learning algorithms. Most publications reflect the use of adaptations to existing learning rules, details on the modifications are highly desired. In conventional computer vision, iterations of training images presented to the network play an important role. Similarly, the biological time of training decides the amount of information provided.

Finally in the testing phase where performance evaluation takes place, specific measurements of SNN models are essential in addition to recognition accuracy. It should include details of the way samples were presented: event rates, and biological time per testing sample. The combination of these two factors determines how much information is presented to the network. An important performance metric is the response time (latency) of an SNN model. A faster model is more suitable for real-time recognition systems such as neuromorphic robotics. A commonly reported characteristic is the accuracy of the network, perhaps adding remarks on how these scores are obtained could help to unify criteria and ease comparison. Work on SNN-based classifications of MNIST are listed in Table 2 and evaluated on the proposed metrics.

Table 3. Hardware dependent comparison

	System	Neuron Model	Synaptic Plasticity	Precision	Simulation Time	Energy/Power Usage
SpiNNaker (Stromatias et al., 2013)	Digital, Scalable	Programmable Neuron/Synapse, Axonal delay	Programmable learning rule	11- to 14-bit synapses	Real-time Flexible time resolution	8 nJ/SE 54.27 MSops/W
TrueNorth (Merolla et al., 2014)	Digital, Scalable	Fixed models, Config params, Axonal delay	No plasticity	122 bits params & states, 4-bit synapse [a]	Real-time	46 GSops/W
Neurogrid (Benjamin et al., 2014)	Mixed-mode, Scalable	Fixed models, Config params	Fixed rule	13-bit shared synapses	Real-time	941 pJ/SE
HI-CANN (Schemmel et al., 2010)	Mixed-mode, Scalable	Fixed models, Config params	Fixed rule	4-bit synapses	Faster than real-time [b]	198 pJ/SE 13.5 MSops/W (network only)
HiAER-IFAT (Yu et al., 2012)	Mixed-mode, Scalable	Fixed models, Config params	No plasticity	Analogue neuron/synapse	Real-time	22-pJ/SE (Park et al., 2014) 20GSops/W

[a] We consider them 4-bit synapses because it is only possible to choose between 4 different signed integers and whether the synapse is active or not.

[b] A speed-up of up to 10^5 times real time has been reported.

2.4.2 Hardware-Specific Evaluation Depending on how neurons, synapses and spike transmission are implemented neuromorphic systems can be categorised as either analogue, digital, or mixed-mode analogue/digital VLSI circuits. Some analogue implementations exploit sub-threshold transistor dynamics to emulate neurons and synapses directly on hardware (Indiveri et al., 2011) and are more energy-efficient while requiring less area than their digital counterparts (Joubert et al., 2012). However, the behaviour of analogue circuits is largely determined during the fabrication process due to transistor mismatch (Indiveri et al., 2011; Pedram and Nazarian, 2006; Linares-Barranco et al., 2003), while their wiring densities render them impractical for large-scale systems. The majority of mixed-mode analogue/digital neuromorphic platforms, such as the High Input Count Analog Neural Network (HI-CANN) (Schemmel et al., 2010), Neurogrid (Benjamin et al., 2014), HiAER-IFAT (Yu et al., 2012), use analogue circuits to emulate neurons and digital packet-based technology to communicate spikes as AER events. This enables reconfigurable connectivity patterns, while the time of spikes is expressed implicitly since typically a spike reaches its destination in less than a millisecond, thus fulfilling the real-time requirement. Digital neuromorphic platforms such as TrueNorth (Merolla et al., 2014) use digital circuits with finite precision to simulate neurons in an event driven manner to minimise the active power dissipation. Neuromorphic systems suffer from model flexibility, since neurons and synapses are fabricated directly on hardware with only a small subset of parameters exposed to the researcher. SpiNNaker is a biologically inspired, massively-parallel, scalable computing architecture designed by the Advanced Processor Technologies (APT) group at the University of Manchester. SpiNNaker has been optimised to simulate very large-scale spiking neural networks in real-time (Furber et al., 2014). SpiNNaker aims to combine the advantages of conventional computers and neuromorphic hardware by utilising low-power programmable cores and scalable event-driven communications hardware.

A direct comparison between neuromorphic platforms is a non-trivial task due to the different hardware implementation technologies as mentioned above. The metric proposed in Table 3 attempts to expose the advantages and disadvantages of different neuromorphic hardware thus to find out the network properties each platform is suited to. The scalability of a hardware platform determines the network size limit of a neural application running on it. Considering the various neural, synaptic models, plasticity learning rules and lengths of axonal delays, a programmable platform is flexible for diverse SNNs while a hard-wired system supporting only specific models wins for its simpler design and implementation. The classification accuracy of a SNN running on a hardware system can be different from the software simulation, since hardware implementation limits on the precision used for the membrane potential of neurons (for the digital platforms) and the synaptic weights. Thus comparison metrics is supposed to include precision as a major assessment of the system performance. Simulation time is another important measure of running large-scale networks on hardware. Real-time implementation is an essential requirement for robotic systems because of the real-time input from the neuromorphic sensors. Running faster than real time is attractive for large/long simulations. However, due to the limitation of hardware resources simulation time may accelerate or slow down according to the network topology and spike dynamics. Also finer time resolution plays an important role in precision sensitive neural models or in sub-millisecond tasks (Lagorce et al., 2015). Comparing the performance of each platform in terms of energy requirements is an interesting comparison metric especially if targeted for mobile applications and robotics. Some researchers have suggested the use of energy per synaptic event (J/SE) (Sharp et al., 2012; Stromatias et al., 2013) as an energy metric because the large fan in and out of a neuron tend to dominate the total energy dissipation during a simulation. Merolla et al. proposed the number of synaptic operations per Watt (Sops/W) (Merolla et al., 2014). These two measurements are the same presentations of energy use of synaptic events, since J/SE×Sops/W = 1 s.

For a particular SNN application or benchmark, the scalability and programmability will determine whether the network is able to run on a platform. The system performance will be assessed on the accuracy, simulation time and energy use running the network. Table 3 aims to summarise the aforementioned hardware comparison metrics.

3 RESULTS

In this section, we present two recognition SNN models working on the Poissonian subset of the NE15-MNIST dataset. Their network components, training and testing methods are described according to the evaluation methodology stated above. The specific spike-based evaluations on input event rates and/or responding latency are also provided. Meanwhile, as tentative benchmarks the models are implemented on SpiNNaker to assess the performance against software simulators. Presenting proper benchmarks for vision recognition systems is still under investigation, the case studies only make first attempt.

3.1 CASE STUDY I

The first case study is a simple two-layered network where the input neurons receive Poissonian presented spike trains from the dataset and form a fully connected network with the decision neurons. There is at least one decision neuron per digit to classify a test input. The neuron with highest output firing rate classifies a test image into the digit it represents. The model utilises LIF neurons, and the parameters are all with biological means, see the listed values in Table 4. The LIF neuron model follows the membrane potential dynamics:

$$\tau_m \frac{\mathrm{d}V}{\mathrm{d}t} = V_{rest} - V + R_m I_{syn}(t) \quad , \tag{2}$$

where τ_m is the membrane time constant, V_{rest} is the resting potential, R_m is the membrane resistance and I_{syn} is the synaptic input current. In PyNN, R_m is presented by $R_m = \tau_m / C_m$, where C_m is the membrane capacitance. A spike is generated when the membrane potential goes beyond the threshold, V_{thresh} and the membrane potential resets to V_{reset}. In addition, a neuron cannot fire within the refractory period, τ_{refrac}, after generating a spike.

The connections between the input neurons and the decision neurons are plastic, so the connection weights can be modulated during training with a standard STDP learning rule. The model is described with PyNN and the code is published in the same Github repository with the dataset. As a potential benchmark, this system is composed with simple neural models, trained with standard learning rules and written in a unified SNN description language. These characteristics allow the same network to be tested on various simulators, both software- and hardware-based.

Both the training and testing exploit the Poissonian subset of the NE15-MNIST dataset. This makes performance evaluation on different simulators possible with the unified spike source array provided by the dataset. In terms of this case study, the performance of the model was evaluated with both software simulation [on NEST (Gewaltig and Diesmann, 2007)] and hardware implementation (on SpiNNaker).

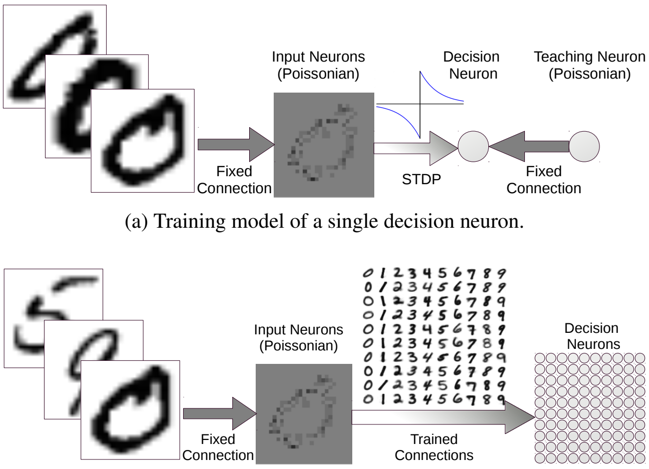
In order to fully assess the performance, different settings have been configured on the network, such as network size, input rate and testing images duration. For simplicity of describing the system, one standard configuration is set as the example in the following sections.

Table 4. Parameter setting for the current-based LIF neurons using PyNN.

Parameters	Values	Units
cm	0.25	nF
tau_m	20.0	ms
tau_refrac	2.0	ms
v_reset	-70.0	mV
v_rest	-65.0	mV
v_thresh	-50.0	mV

3.1.1 Training There are two layers in the model: 28×28 input neurons fully connect to 100 decision neurons. Each decision neuron responds to a certain template of a digit. In the standard configuration, there are 10 decision neurons answering to the same digit with slightly different templates. Those templates are embedded in the connection weights between the two layers. Figure 6(a) shows how the connections to a single decision neuron are tuned.

The training set of 60,000 hand written digits are firstly classified into 100 classes, 10 subclasses per digit, using K-means clusters. K-means clustering separates a set of data points into K subsets (clusters) according to the Euclidean distance. Therefore each cluster tends to form a boundary within which the data points are near each other. In this case, all the images of a same digit (a class) are divided into 10 subclasses by assigning K=10. Then the images in a certain subclass are used to train a template embedded in the synaptic weights to the corresponding decision neuron. The firing rates of the input neurons are assigned linearly according to their intensities and the total firing rate of all the 28×28 input neurons is normalised with 2,000 Hz, e.g. the summation of the firing rate of all the input neurons is 2,000 Hz. All the images together are presented for 18,000 s (about 300 ms per image) during training and at the same time a teaching signal of 50 Hz is conveyed to the decision neuron to trigger STDP learning. The trained weights are plotted in accordance with the positions of the decision neurons in Figure 6(b).



(a) Training model of a single decision neuron.

(b) Testing model of the spiking neural network.

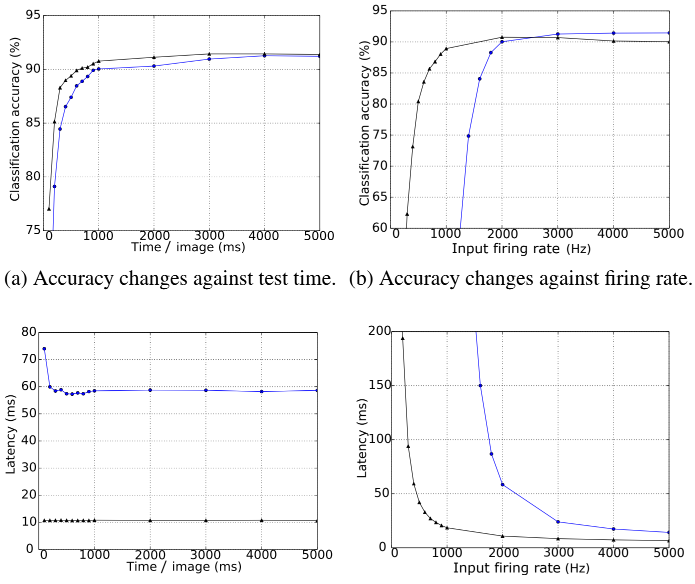
Figure 6: The training and testing model of the two-layered spiking neural network.

3.1.2 Testing After training the weights of the plastic synapses are set to static, keeping the state of the weights at the last moment of training. The weak weights were set to inhibitory connections with an identical strength. The feed-forward testing network is shown in Figure 6(b) where Poissonion spike trains are generated the same way as in the training with a total firing rate of 2,000 Hz per image. The input neurons convey the same spike trains to every decision neuron through its responding trained synaptic weights. Every testing image (10,000 images in total) is presented once and lasts 1 s with a silence of 200 ms between them. The output neuron with the highest firing rate decides what digit was recognized. Taken the trained weights from the NEST simulation, the accuracy of the recognition on NEST reaches 90.03% with the standard configuration, while the result drops slightly to 89.97% using SpiNNaker. In comparison, both trained and tested on SpiNNaker the recognition accuracy is 87.41%, and with the same weights applied to NEST the result turns out to be 87.25%.

3.1.3 Evaluation The evaluation starts from the hardware-independent side, focusing on the spike-based recognition analysis. As mentioned in Section 2.4.1, CA and response time (latency) are the main concerns when assessing the recognition capability. In our experiment, two sets of weights were applied: the original STDP trained weights and scaled-up weights which are 10 times stronger. The spiking rates of the testing samples were also modified, ranging from 10 to 5,000 Hz.

We found that accuracy depends largely on the time each sample is exposed to the network and the sample spiking rate (Figure 7.) Furthermore, the latency of the output of the decision neurons is affected by both the spiking rate and connection weights. Figure 7(a) shows that the CA is better as exposure time increases. The longer an image is presented, the more information is gathered by the network, so the accuracy climbs. Classification accuracy also increases when input spiking rates are augmented (Figure 7(b)). Given that the spike trains injected into the network are more intense, the decision neurons become more active and so does the output disparity among them. Nonetheless, it is important to know that these increments in CA have a limit, as is shown in the aforementioned figures. With stronger weights, the accuracy is much higher when the input firing rate is less than $2,000$ Hz.

The latency of an SNN model is the result of the input rates and synaptic weights. We calculate the latency of each test by getting the time difference of the first spike generated by the output layer and the first spike of the input layer. As the input rates grow, there are more spikes arriving at the decision neurons, triggering them to spike sooner. A similar idea applies to the influence of synaptic weights. If stronger weights are taken, then the membrane potential of a neuron reaches its threshold earlier. Figure 7(d) indicates that the latency is shortened with increasing input rates with both the original and scaled-up weights. When the spiking rate is less than $2,000$ Hz, the network with stronger weights has a much shorter latency. As long as there are enough spikes to trigger the decision neurons to spike, increasing the test time will not make the network respond sooner (Figure 7(c)).



(a) Accuracy changes against test time. (b) Accuracy changes against firing rate.

(c) Latency stablises against test time. (d) Latency changes against firing rate.

Figure 7: Accuracy and response time (latency) change over test time and input firing rate per test image. The test time is the duration of the presence of a single test image, and the input firing rate is the summation of all the input neurons. Original trained weights are used (circles in blue) as well as the scaled up (×10) weights (triangles in black).

Regarding the network size, it not only influences the accuracy of a model but also the time taken for simulation on specific platforms thus impacting the energy usage on the hardware. For the purpose of comparing the accuracy, simulation time and energy usage, different configurations have been tested on NEST (working on a PC with CPU: i5-4570 and 8G memory) and SpiNNaker, see Table 5. The input rates in all of the tests are $5,000$ Hz, and each image is presented for 1 s. The configurations only differ in the number of templates (subclasses/clusters) per digit. Thus the network size vary according to the number of neurons in the decision layer where each neuron represents a subclass of a digit and the template is embedded in the synaptic weights connecting from the input layer to the decision neuron. The recognition accuracies differ in a range of $\pm 0.5\%$ between NEST and SpiNNaker due to the limited fast memory and the necessity for fixed-point arithmetic on SpiNNaker to ensure real-time operation. It is inevitable that numerical precision will be below IEEE double precision at various points in the processing chain from synaptic input to membrane potential. The main bottleneck is currently in the ring buffer where the total precision for accumulated spike inputs is 16-bit, meaning that individual spikes are realistically going to be limited to 11- to 14-bit depending upon the probabilistic headroom calculated as necessary from the network configuration and spike throughput (Hopkins and Furber, 2015 to be published). As the network size grows there are more decision neurons and synapses connecting to them, thus the simulation time on NEST increases. On the other hand, SpiNNaker works in real (biologically real) time and the simulation time becomes shorter than NEST simulation when 1,000 patterns per digit are used. The Thermal Design Power (TDP) usage of all four processors of i5-4570 actively operating at base frequency is 84 W[2]. NEST was run fully active on a single core which cost 21 W of power usage. The energy use can be calculated as the product of the simulation time and the power use. Even with the smallest network, SpiNNaker wins in the energy cost comparison, see Figure 8. Among different network configurations, the network of 500 decision neurons (50 clusters per digit) reaches the highest recognition rate. The network achieved a CA of 92.98% and average latency of 10.70 ms, and the simulation costs SpiNNaker 0.41 W on power use and 4,920 J on energy use.

Table 5. Comparisons of NEST (N) on a PC and SpiNNaker (S) performances.

Subclasses	Accuracy (%)		Simulation (s)		Power Use (W)	
per digit	N	S	N	S	N	S
1	79.62	79.50	554.77	12,000	21.0	0.38
10	91.29	91.43	621.74			0.38
50	92.98	92.92	1,125.12			0.41
100	87.27	86.83	1,406.01			0.44
1000	89.65	89.74	30,316.88			1.50

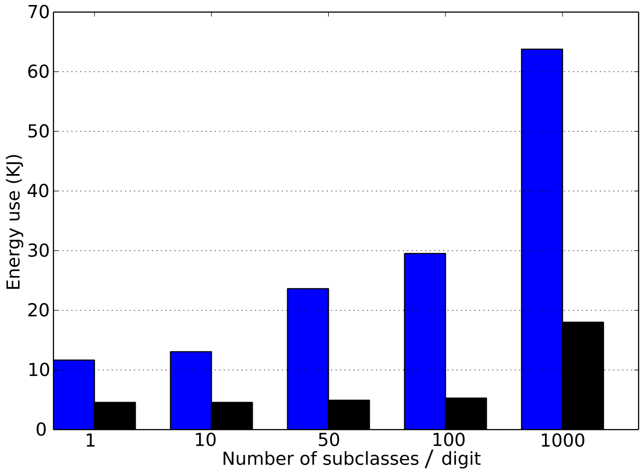


Figure 8: Energy usages of different network size both using NEST (blue) on a PC and SpiNNaker (black).

[2] http://ark.intel.com/products/75043/Intel-Core-i5-4570-Processor-6M-Cache-up-to-3_60-GHz

3.2 CASE STUDY II

Deep learning architectures and in particular Convolutional Networks (LeCun et al., 1998) and Deep Belief Networks (DBNs) (Hinton et al., 2006) have been characterised as one of the breakthrough technologies of the decade (Hof, 2013). One of the advantages of these type of networks is that their performance can be increased by adding more layers (Hinton et al., 2006).

However, state-of-the-art deep networks comprise a large number of layers, neurons and connections resulting in high energy demands, communication overheads, and high response latencies. This is a problem for mobile and robotic platforms which may have limited computational and power resources but require fast system responses.

O'Connor et al. (2013) proposed a method to map off-line trained DBNs into a spiking neural networks and take advantage of the real-time performance and energy efficiency of neuromorphic platforms. This led initially to an implementation on an event-driven Field-Programmable Gate Array (FPGA) called Minitaur (Neil and Liu, 2014) and then on the SpiNNaker platform (Stromatias et al., 2015a). For this work we used an off-line trained[3] spiking DBN with a 784-500-500-10 network topology. Simulations take place on a software spiking neural network simulator named Brian (Goodman and Brette, 2008) and results are verified on the SpiNNaker platform.

3.2.1 Training DBNs consist of stacked Restricted Boltzmann Machines (RBMs), which are fully connected recurrent networks but without any connections between neurons of the same layer. Training is performed unsupervised using the standard CD rule (Hinton et al., 2006) and only the output layer is trained in a supervised manner. The main difference between spiking DBNs and traditional DBNs is the activation function used for the neurons. O'Connor et al. (2013) proposed the use of the Siegert approximation (Jug et al., 2012) as the activation function, which returns the expected firing rate of a LIF neuron given input firing rates, input weights, and standard neuron parameters.

3.2.2 Testing After the training process the learnt synaptic weights can be used in a spiking neural network which consists of LIF neurons with delta-current synapses. Table 6 shows the LIF parameters used in the simulations.

Table 6. Default parameters of the Leaky Integrate-and-Fire Model used in simulations.

Parameters	Values	Units
tau_m	5	s
tau_refrac	2.0	ms
v_reset	0.0	mV
v_rest	0.0	mV
v_thresh	1.0	mV

The pixels of each MNIST digit from the testing set are converted into Poisson spike trains with a rate proportional to the intensity of their pixel, while their firing rates are scaled so that the total firing rate of the input population is constant (O'Connor et al., 2013).

The CA was chosen as the performance metric of the spiking DBN, which is the percentage of the correctly classified digits over the whole MNIST testing set.

3.2.3 Evaluation Neuromorphic platforms may have limited hardware resources to store the synaptic weights (Schemmel et al., 2010; Merolla et al., 2014). In order to investigate how the precision of the

[3] https://github.com/dannyneil/edbn/

weights affects the CA of a spiking DBN the double floating point weights of the offline trained network were converted to different fixed-point representations. The following notation will be used throughout this paper, Q*m*.*f*, where *m* signifies the number of bits for the integer part (including the sign bit) and *f* the number of bits used for the fractional part.

Figure 9 shows the effect of reduced weight bit precision on the CA for different input firing rates on the Brian simulator. Using the same weight precision of Q3.8, SpiNNaker achieved a CA of 94.94% when 1,500 Hz was used for the input population (Stromatias et al., 2015a). Brian for the same firing rates and weight precision achieved a CA of 94.955%. Results are summarised in Table 7. The slightly lower CA of the SpiNNaker simulation indicates that not only the weight precision but also the precision of the membrane potential affects the overall classification performance.

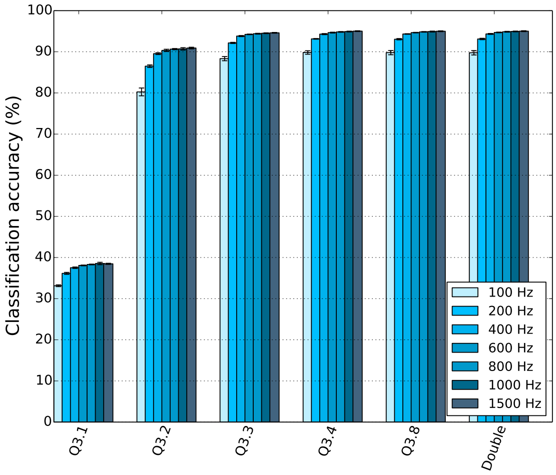


Figure 9: CA as a function of the weight bit precision for different input firing rates.

Table 7. Classification accuracy (CA) of the same DBN running on different platforms.

Simulator	CA (%)	Weight Precision
Matlab	96.06	Double floating point
Brian	95.00	Double floating point
Brian	94.955	Q3.8
SpiNNaker	94.94	Q3.8

Stromatias et al. (2015b) showed that spiking DBNs are capable of maintaining a high CA even for weight precisions down to Q3.3, while they are also remarkably robust to high levels of input noise regardless of the weight precision.

A similar experiment to the one presented for the Case Study I was performed; its purpose was to establish the relation that input spike rates hold with latency and classification accuracy. The input rates were varied from 500 Hz to 2,000 Hz and the results are summarised in Figure 10. Simulations ran in Brian for all 10,000 MNIST digits of the testing set and for 4 trials. Figure 11 shows a histogram of the classification latencies on SpiNNaker when the input rates are 1,500 Hz. The mean classification latency for the particular spiking DBN on SpiNNaker is 16 ms which is identical to the Brian simulation seen in Figure 10.

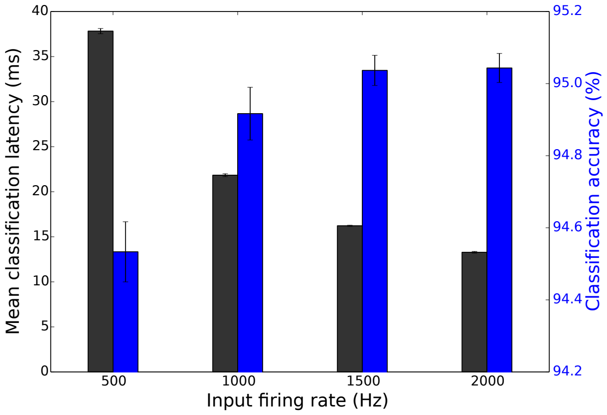


Figure 10: Mean classification latency (black) and classification accuracy (blue) as a function of the input firing rate for the spiking DBN. Results are averaged over 4 trials, error bars show standard deviations.

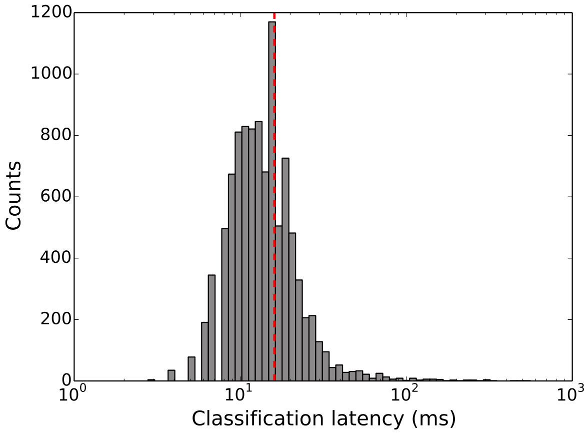


Figure 11: Histogram of the classification latencies for the MNIST digits of the testing set when the input rates are set to 1,500 Hz. The mean classification latency of the spiking DBN on SpiNNaker is 16 ms.

Finally, this particular spiking DBN ran on a single SpiNNaker chip (16 ARM9 cores) and dissipated less than 0.3 W when 2,000 spikes per second per digit were used, as seen in Figure 12. The identical network ran on Minitaur (Neil and Liu, 2014), an event-driven FPGA implementation, and consumed 1.5 W when 1,000 spikes per image were used.

4 DISCUSSION

4.1 SUMMERY OF THE WORK

This paper puts forward the NE dataset as a baseline for comparisons on vision based SNNs. It contains converted spike representations of existing widely-used databases in the vision recognition field. Since new problems will be introduced continuously before vision becomes a solved question, the dataset will

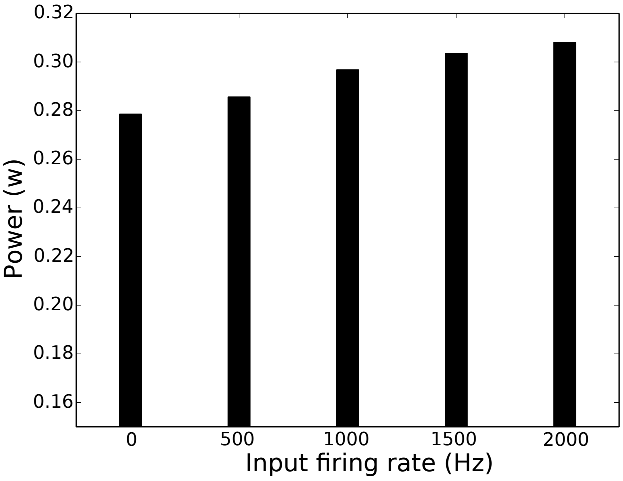


Figure 12: Power dissipation of a spiking DBN running on a single SpiNNaker chip as a function of the total input firing rate.

evolve as research develops. The conversion methods transforming images and videos to spike trains will advance. The number of vision databases included will increase and the corresponding evaluation methodology will evolve as well. The dataset aims to provide a unified spike-based vision database and complementary evaluation methodologies to assess the performance of various SNN algorithms.

The first launch of the dataset is published as NE15-MNIST, which contains four different spike presentations of the stationary hand-written digit database. The Poissonian subset aims at benchmarking the existing rate-based recognition methods. The rank-order-encoded subset, FoCal, encourages research into spatio-temporal algorithms on recognition applications using only small numbers of input spikes. Fast recognition can be verified on the subset of DVS recorded flashing input, since merely 30 ms of useful spike trains are recorded for each image. As a step forward, the continuous spike trains captured from the DVS recorded moving input can be a good test on mobile neuromorphic robots.

The complementary evaluation methodology is essential to assess both the model-level and hardware-level performances. For a network model, its topology, neuron and synapse models, and training methods are major descriptions for any kind of neural networks, including SNNs. While the recognition accuracy, network latency and also the biological time taken for both training and testing are specific performance measurements of a spike-based model. To build any SNN model on a hardware platform, its network size will be constrained by the scalability of the hardware. Neural and synaptic models are limited to the ones that are physically implemented, unless the hardware platform supports programmability. The accuracy of the results (e.g. CA) are naturally affected by the precision of the variable representing the membrane potential and synaptic weights. Any attempt to implement an on-line learning algorithm on neuromorphic hardware must be backed by synaptic plasticity support. Running an identical SNN model on different neuromorphic hardware platforms can not only expose if any of the previously mentioned capacities are supported, but also benchmark their performance on simulation time and energy usage.

Using the Poissonian subset of the NE15-MNIST dataset, two benchmark systems were proposed. The models were described and their performance on accuracy, network latency, simulation time and energy usage were presented. These example benchmarking systems provided a recommended way of using the dataset and evaluating system performance. They provide a baseline for comparisons and encourage improved algorithms and models to make use of the dataset.

Although spike-based algorithms have not surpassed their non-spiking counterparts in terms of recognition accuracy, they have shown great performance in response time and energy efficiency.

The dataset makes the comparison of SNNs with conventional recognition methods possible by using converted spike presentations of the same vision databases. As far as we know, it is the first attempt to benchmarking neuromorphic vision recognition by providing public spike-based dataset and the evaluation metrics. As the dataset grows, it will allow new problems to be investigated by researchers, which should allow the identification of future directions and, in consequence, advance the field.

4.2 THE FUTURE DIRECTION OF AN EVOLVING DATABASE

The database will expand by converting more popular vision datasets to spike representations. As mentioned in Section 1, face recognition has become a hot topic in SNN approaches, however there is no unified spike-based dataset to benchmark theses networks. Thus, the next development step for our dataset is to include face recognition databases. While viewing an image, saccades direct high-acuity visual analysis to a particular object or a region of interest and useful information is gathered during the fixation of several saccades in a second. It is possible to measure the scan path or trajectory of the eyeball and the trajectories showed particular interest in eyes, nose and mouth while viewing a human face (Yarbus, 1967). Therefore, our plan is also to embed modulated trajectory information to direct the recording using DVS sensors to simulate human saccades.

There will be more methods and algorithms of converting images to spikes. Although Poisson spikes are the most commonly used external input of an SNN system, there are several *in-vivo* recordings in different cortical areas showing that the inter-spike intervals (ISI) are not Poissonian (Deger et al., 2012). Thus Deger et al. (2012) proposed new algorithms to generate superposition spike trains of Poisson process with dead-time (PPD) and of Gamma process. Including novel spike generation algorithms to the dataset is one of the future work to satisfy the large varieties of research.

Each encounter of an object on the retina is completely unique, because of the illumination (lighting conditions), position (projection locations on the retina), scale (distances and sizes), pose (viewing angles), and clutter (visual contexts) variabilities. But the brain recognises a huge number of objects rapidly and effortlessly even in cluttered and natural scenes. In order to explore invariant object recognition, the dataset is going to include the NORB (NYU Object Recognition Benchmark) dataset (LeCun et al., 2004), which contains images of objects that are first photographed in ideal conditions and then moved and placed in front of natural scene images.

Action recognition will be the first problem of video processing to be introduced in the dataset. The initial plan is to use the DVS retina to convert KTH and Weizmann benchmarks to spiking versions. Meanwhile, providing a software DVS retina simulator to transform frames into spike trains is also on the schedule. By doing this, huge number of videos, such as in YouTube, can automatically be converted to spikes, therefore providing researchers more time to work on their own applications.

In all, it is impossible to provide enough datasets, converting methods or benchmarking results by the dataset proposers ourselves, thus we encourage researchers contribute to the dataset. In another word, people could public their data in the dataset allowing future comparisons on the same data source; researchers could share the spike converting algorithms by generating dataset to promote the corresponding recognition methods; and neuromorphic hardware owners are welcome to provide benchmarking results to compare the system performance.

ACKNOWLEDGMENTS

The work presented in this paper was largely inspired by discussions at the 2015 Workshops on Neuromorphic Cognition Engineering in CapoCaccia. The authors would like to thank the organisers and the sponsors. The authors would also like to thank Patrick Camilleri, Michael Hopkins, and John Woods for the meaningful discussions and proofreading of the paper. The construction of the SpiNNaker machine was supported by the Engineering and Physical Science Research Council (EPSRC grant EP/4015740/1) with additional support from industry partners ARM Ltd and Silistix Ltd. The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement n. 320689 and also from the EU Flagship Human Brain Project (FP7-604102).

REFERENCES

Benjamin, B. V., Gao, P., McQuinn, E., Choudhary, S., Chandrasekaran, A. R., Bussat, J.-M., et al. (2014). Neurogrid: a mixed-analog-digital multichip system for large-scale neural simulations. *Proceedings of the IEEE* 102, 699–716

Beyeler, M., Dutt, N. D., and Krichmar, J. L. (2013). Categorization and decision-making in a neurobiologically plausible spiking network using a STDP-like learning rule. *Neural Networks* 48, 109–124

Bichler, O., Querlioz, D., Thorpe, S. J., Bourgoin, J.-P., and Gamrat, C. (2012). Extraction of temporally correlated features from dynamic vision sensors with spike-timing-dependent plasticity. *Neural Networks* 32, 339–348

Bill, J. and Legenstein, R. (2014). A compound memristive synapse model for statistical learning through STDP in spiking neural networks. *Frontiers in neuroscience* 8

Blank, M., Gorelick, L., Shechtman, E., Irani, M., and Basri, R. (2005). Actions as space-time shapes. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*. vol. 2, 1395–1402

Brader, J. M., Senn, W., and Fusi, S. (2007). Learning real-world stimuli in a neural network with spike-driven synaptic dynamics. *Neural computation* 19, 2881–2912

Camunas-Mesa, L., Zamarreño-Ramos, C., Linares-Barranco, A., Acosta-Jiménez, A. J., Serrano-Gotarredona, T., and Linares-Barranco, B. (2012). An event-driven multi-kernel convolution processor module for event-driven vision sensors. *Solid-State Circuits, IEEE Journal of* 47, 504–517

Cao, Y., Chen, Y., and Khosla, D. (2015). Spiking deep convolutional neural networks for energy-efficient object recognition. *International Journal of Computer Vision* 113, 54–66

Davison, A. P., Brüderle, D., Eppler, J., Kremkow, J., Muller, E., Pecevski, D., et al. (2008). PyNN: a common interface for neuronal network simulators. *Frontiers in neuroinformatics* 2

Deger, M., Helias, M., Boucsein, C., and Rotter, S. (2012). Statistical properties of superimposed stationary spike trains. *Journal of computational neuroscience* 32, 443–463

Delbruck, T. (2008). Frame-free dynamic digital vision. In *Proceedings of Intl. Symp. on Secure-Life Electronics, Advanced Electronics for Quality Life and Society*. 21–26

Delorme, A. and Thorpe, S. J. (2001). Face identification using one spike per neuron: resistance to image degradations. *Neural Networks* 14, 795–803

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: a large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. 248–255

DiCarlo, J. J., Zoccolan, D., and Rust, N. C. (2012). How does the brain solve visual object recognition? *Neuron* 73, 415–434

Diehl, P., Neil, D., Binas, J., Cook, M., Liu, S.-C., and Pfeiffer, M. (2015). Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In *Neural Networks (IJCNN), The 2015 International Joint Conference on*. to be published

Diehl, P. U. and Cook, M. (2015). Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in Computational Neuroscience* 9, 99

Eliasmith, C., Stewart, T. C., Choo, X., Bekolay, T., DeWolf, T., Tang, Y., et al. (2012). A large-scale model of the functioning brain. *science* 338, 1202–1205

Felleman, D. J. and Van Essen, D. C. (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cerebral cortex* 1, 1–47

Fu, S.-Y., Yang, G.-S., and Kuai, X.-K. (2012). A spiking neural network based cortex-like mechanism and application to facial expression recognition. *Computational intelligence and neuroscience* 2012, 19

Furber, S. and Temple, S. (2007). Neural systems engineering. *Journal of the Royal Society interface* 4, 193–206

Furber, S. B., Galluppi, F., Temple, S., Plana, L., et al. (2014). The SpiNNaker Project. *Proceedings of the IEEE* 102, 652–665

Gewaltig, M.-O. and Diesmann, M. (2007). NEST (NEural Simulation Tool). *Scholarpedia* 2, 1430

Goodman, D. and Brette, R. (2008). Brian: a simulator for spiking neural networks in Python. *Frontiers in neuroinformatics* 2

Hinton, G. E., Osindero, S., and Teh, Y.-W. (2006). A fast learning algorithm for Deep Belief Nets. *Neural computation* 18, 1527–1554

Hof, R. (2013). 10 breakthrough technologies 2013

Hopkins, M. and Furber, S. (2015 to be published). Accuracy and Efficiency in Fixed-Point Neural ODE Solvers. *Neural computation*

Indiveri, G., Linares-Barranco, B., Hamilton, T. J., Van Schaik, A., Etienne-Cummings, R., Delbruck, T., et al. (2011). Neuromorphic silicon neuron circuits. *Frontiers in neuroscience* 5

Joubert, A., Belhadj, B., Temam, O., and Héliot, R. (2012). Hardware spiking neurons design: analog or digital? In *Neural Networks (IJCNN), The 2012 International Joint Conference on* (IEEE), 1–5

Jug, F., Lengler, J., Krautz, C., and Steger, A. (2012). Spiking networks and their rate-based equivalents: does it make sense to use Siegert neurons? In *Swiss Soc. for Neuroscience*

Kolb, H. (2003). How the retina works. *American scientist* 91, 28–35

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105

Lagorce, X., Stromatias, E., Galluppi, F., Plana, L. A., Liu, S.-C., Furber, S. B., et al. (2015). Breaking the millisecond barrier on SpiNNaker: implementing asynchronous event-based plastic models with microsecond resolution. *Frontiers in Neuroscience* 9, 206

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86, 2278–2324

LeCun, Y., Huang, F. J., and Bottou, L. (2004). Learning methods for generic object recognition with invariance to pose and lighting. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*. vol. 2, II–97

Linares-Barranco, B., Serrano-Gotarredona, T., and Serrano-Gotarredona, R. (2003). Compact low-power calibration mini-DACs for neural arrays with programmable weights. *Neural Networks, IEEE Transactions on* 14, 1207–1216

Liu, J., Luo, J., and Shah, M. (2009). Recognizing realistic actions from videos "in the wild". In *Computer Vision and Pattern Recognition, 2009. CVPR. IEEE Conference on*. 1996–2003

Liu, Q. and Furber, S. (2015). Real-time recognition of dynamic hand postures on a neuromorphic system. In *Artificial Neural Networks, 2015. ICANN. International Conference on*. vol. 1, 979

Lyons, M., Akamatsu, S., Kamachi, M., and Gyoba, J. (1998). Coding facial expressions with gabor wavelets. In *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*. 200–205

Mahowald, M. (1992). *"VLSI analogs of neuronal visual processing: a synthesis of form and function"*. Ph.D. thesis, California Institute of Technology

Masmoudi, K., Antonini, M., Kornprobst, P., and Perrinet, L. (2010). A novel bio-inspired static image compression scheme for noisy data transmission over low-bandwidth channels. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. 3506–3509

Matsugu, M., Mori, K., Ishii, M., and Mitarai, Y. (2002). Convolutional spiking neural network model for robust face detection. In *Neural Information Processing, 2002. ICONIP'02. Proceedings of the 9th International Conference on*. vol. 2, 660–664

Merolla, P. A., Arthur, J. V., Alvarez-Icaza, R., Cassidy, A. S., Sawada, J., Akopyan, F., et al. (2014). A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* 345, 668–673

Neftci, E., Das, S., Pedroni, B., Kreutz-Delgado, K., and Cauwenberghs, G. (2013). Event-driven contrastive divergence for spiking neuromorphic systems. *Frontiers in neuroscience* 7

Neil, D. and Liu, S.-C. (2014). Minitaur, an event-driven FPGA-based spiking network accelerator. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on* 22, 2621–2628

Nessler, B., Pfeiffer, M., Buesing, L., and Maass, W. (2013). Bayesian computation emerges in generic cortical microcircuits through spike-timing-dependent plasticity. *PLoS Comput Biol*

O'Connor, P., Neil, D., Liu, S.-C., Delbruck, T., and Pfeiffer, M. (2013). Real-time classification and sensor fusion with a spiking deep belief network. *Frontiers in neuroscience* 7

Park, J., Ha, S., Yu, T., Neftci, E., and Cauwenberghs, G. (2014). "a 65k-neuron 73-mevents/s 22-pj/event asynchronous micro-pipelined integrate-and-fire array transceiver". In *Biomedical Circuits and Systems Conference (BioCAS), 2014 IEEE* (IEEE), 675–678

Pedram, M. and Nazarian, S. (2006). Thermal modeling, analysis, and management in VLSI circuits: principles and methods. *Proceedings of the IEEE* 94, 1487–1501

Posch, C., Serrano-Gotarredona, T., Linares-Barranco, B., and Delbruck, T. (2014). Retinomorphic event-based vision sensors: bioinspired cameras with spiking output. *Proceedings of the IEEE* 102, 1470–1484

Riesenhuber, M. and Poggio, T. (1999). Hierarchical models of object recognition in cortex. *Nature neuroscience* 2, 1019–1025

Schemmel, J., Bruderle, D., Grubl, A., Hock, M., Meier, K., and Millner, S. (2010). A wafer-scale neuromorphic hardware system for large-scale neural modeling. In *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*. 1947–1950

Schüldt, C., Laptev, I., and Caputo, B. (2004). Recognizing human actions: a local SVM approach. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on* (IEEE), vol. 3, 32–36

Sen, B. and Furber, S. (2009). Evaluating rank-order code performance using a biologically-derived retinal model. In *Neural Networks, 2009. IJCNN. International Joint Conference on* (IEEE), 2867–2874

Serrano-Gotarredona, T. and Linares-Barranco, B. (2013). A 128×128 1.5% contrast sensitivity 0.9% FPN 3μs latency 4 mW asynchronous frame-free dynamic vision sensor using transimpedance preamplifiers. *Solid-State Circuits, IEEE Journal of* 48, 827–838

Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., and Poggio, T. (2007). Robust object recognition with cortex-like mechanisms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 29, 411–426

Sharp, T., Galluppi, F., Rast, A., and Furber, S. (2012). Power-efficient simulation of detailed cortical microcircuits on SpiNNaker. *Journal of neuroscience methods* 210, 110–118

Squire, L. R. and Kosslyn, S. M. (1998). *Findings and current opinion in cognitive neuroscience* (MIT Press)

Stromatias, E., Galluppi, F., Patterson, C., and Furber, S. (2013). Power analysis of large-scale, real-time neural networks on SpiNNaker. In *Neural Networks (IJCNN), The 2013 International Joint Conference on*. 1–8

Stromatias, E., Neil, D., Galluppi, F., Pfeiffer, M., Liu, S.-C., and Furber, S. (2015a). Scalable energy-efficient, low-latency implementations of trained spiking deep belief networks on SpiNNaker. In *Neural Networks (IJCNN), The 2015 International Joint Conference on* (IEEE), to be published

Stromatias, E., Neil, D., Pfeiffer, M., Galluppi, F., Furber, S. B., and Liu, S.-C. (2015b). Robustness of spiking deep belief networks to noise and reduced bit precision of neuro-inspired hardware platforms. *Frontiers in neuroscience* 9

Van Rullen, R. and Thorpe, S. J. (2001). Rate coding versus temporal order coding: what the retinal ganglion cells tell the visual cortex. *Neural computation* 13, 1255–1283

Van Rullen, R. and Thorpe, S. J. (2002). Surfing a spike wave down the ventral stream. *Vision research* 42, 2593–2615

Yang, M., Liu, S.-C., and Delbruck, T. (2015). A dynamic vision sensor with 1% temporal contrast sensitivity and in-pixel asynchronous delta modulator for event encoding. *Solid-State Circuits, IEEE Journal of* 50, 2149–2160

Yarbus, A. L. (1967). *Eye movements during perception of complex objects* (Springer)

Yu, T., Park, J., Joshi, S., Maier, C., and Cauwenberghs, G. (2012). 65k-neuron Integrate-and-Fire array transceiver with address-event reconfigurable synaptic routing. In *Biomedical Circuits and Systems Conference (BioCAS), 2012 IEEE*. 21–24

Zhao, B., Ding, R., Chen, S., Linares-Barranco, B., and Tang, H. (2015). Feedforward categorization on AER motion events using cortex-like features in a spiking neural network. *Neural Networks and Learning Systems, IEEE Transactions on* 26, 1963–1978