

Benchmarking Spike-Based Visual Recognition: a Dataset and Evaluation

Qian Liu^{1*}, Garibaldi Pineda-Garcia¹, Evangelos Stamatias¹, Teresa Serrano-Gotarredona², Steve B. Furber¹

¹School of Computer Science, University of Manchester, United Kingdom, ²Instituto de Microelectronica de Sevilla, Spain

Submitted to Journal:
Frontiers in Neuroscience

Specialty Section:
Neuromorphic Engineering

Article type:
Original Research Article

Manuscript ID:
169532

Received on:
18 Sep 2015

Frontiers website link:
www.frontiersin.org

Conflict of interest statement

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest

Author contribution Statement

QL: Main author, leading the writing and experiments.

GP: Main writer and main contributor to the dataset.

ES: Main writer on the experiments.

TS: Contributor to the dataset.

SF: Theoretical contributor

Keywords

Benchmarking, Vision Dataset, Evaluation, neuromorphic engineering, spiking neural networks

Abstract

Word count: 310

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 neuromorphic 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.



1

Benchmarking Spike-Based Visual Recognition: a Dataset and Evaluation

Qian Liu^{1,*}, Garibaldi Pineda-García¹, Evangelos Stamatias¹,
Teresa Serrano-Gotarredona², and Steve Furber¹

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

2 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 neuromorphic 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,

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

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

1 INTRODUCTION

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

50 Riesenhuber and Poggio (1999) proposed a quantitative modelling framework of object recognition
51 with position-, scale- and view-invariance based on the units of MAX-like operations. The cortical-like
52 model has been analysed on several datasets (Serre et al., 2007). And recently Fu et al. (2012) reported
53 that their SNN implementation of the framework was capable of facial expression recognition with a
54 classification accuracy (CA) of 97.35% on the JAFFE dataset (Lyons et al., 1998) which contains 213
55 images of 7 facial expressions posed by 10 individuals. They employed simple integrate-and-fire neurons
56 with rank order coding (ROC) where the earliest pre-synaptic spikes have the strongest impact on the post
57 synaptic potentials. According to Van Rullen and Thorpe (2002), the first wave of spikes carry explicit
58 information through the ventral stream and in each stage meaningful information is extracted and spikes
59 are regenerated. Using one spike per neuron, Delorme and Thorpe (2001) reported 100% and 97.5%
60 accuracies on the face identification task over changing contrast and luminance training (40 individuals ×
61 8 images) and testing data (40 individuals × 2 images) respectively.

62 The Convolutional Neural Network (CNN), also known as the *ConvNet* developed by LeCun et al.
63 (1998), is a well applied model of such a cortex-like framework. An early Convolutional Spiking Neural
64 Network (CSNN) model identified faces of 35 persons with a CA of 98.3% exploiting simple integrate
65 and fire neurons (Matsugu et al., 2002). Another CSNN model (Zhao et al., 2015) was trained and tested
66 both with DVS raw data and Leaky Integrate-and-Fire (LIF) neurons. It was capable of recognising three
67 moving postures with a CA of about 99.48% and 88.14% on the MNIST-DVS dataset (see Chapter 4). As
68 one step forward, Camunas-Mesa et al. (2012) implemented a convolution processor module in hardware
69 which could be combined with a DVS for high-speed recognition tasks. The inputs of the ConvNet were
70 continuous spike events instead of static images or frame-based videos. The chip detected four suits of a
71 52 card deck while the cards were fast browsed in only 410 ms. Similarly, a real-time gesture recognition
72 model (Liu and Furber, 2015) was implemented on a neuromorphic system with a DVS as a front-end
73 and a SpiNNaker (Furber et al., 2014) machine as the back-end where LIF neurons built up the ConvNet
74 configured with biological parameters. In this study’s largest configuration, a network of 74,210 neurons
75 and 15,216,512 synapses used 290 SpiNNaker cores in parallel and reached 93.0% accuracy.

76 Deep Neural Networks (DNNs) together with deep learning are the most exciting research fields in
77 vision recognition. The spiking deep network has great potential to combine remarkable performance

78 with the energy efficient training and running. In the initial stage of the research, the study was focused
79 on converting off-line trained deep network to SNNs (O'Connor et al., 2013). The same network initially
80 implemented on a FPGA achieved a CA of 92.0% (Neil and Liu, 2014), while a later implementation on
81 SpiNNaker scored 95.0% (Stromatias et al., 2015a). Recent attempts have contributed to better translation
82 by utilising modified units in a ConvNet (Cao et al., 2015) and tuning the weights and thresholds (Diehl
83 et al., 2015)). The later paper claims a state-of-the-art performance (99.1% on the MNIST dataset) compari-
84 ring to original ConvNet. The current trend of training Spiking DNNs on-line using biologically-plausible
85 learning methods is also promising. An event driven Contrastive Divergence (CD) training algorithm for
86 RBMs (Restricted Boltzmann Machines) was proposed for Deep Belief Networks (DBN) using LIF neu-
87 rons with STDP (Spike-Timing-Dependent Plasticity) synapses and verified on MNIST (91.9%) (Neftci
88 et al., 2013).

89 STDP as a biological learning process is applied to vision tasks. Bichler et al. (2012) demonstrated
90 an unsupervised STDP learning model to classify car trajectories captured with a DVS retina. A similar
91 model was tested on a Poissonian spike presentation of the MNIST dataset achieving a performance of
92 95.0% (Diehl and Cook, 2015). Theoretical analysis (Nessler et al., 2013) showed that unsupervised STDP
93 was able to approximate a stochastic version of Expectation Maximization, a powerful learning algorithm
94 in machine learning. The computer simulation achieved 93.3% CA on MNIST and could be implemented
95 in a memristive device (Bill and Legenstein, 2014).

96 Despite the promising research on SNN-based vision recognition, there is no commonly used database
97 in the format of spikes. In the studies listed above, all the vision data used are in one of the following
98 formats: (1) the grey-scale raw values of images; (2) rate-based spike trains according to pixel intensities
99 created by various Poissonian generators; (3) unpublished DVS recorded spike-based videos. As a con-
100 sequence, a new series of spike-based vision datasets is now needed to quantitatively measure progress
101 within this rapidly advancing field and to provide fair competition resources for researchers. Apart from
102 using spikes instead of the frame-based data of conventional computer vision, there are new concerns of
103 evaluating neuromorphic vision in tasks other than recognition accuracy. Therefore a common metric of
104 performance evaluation on spike-based vision is also required to specify the measurements of algorithms
105 and models. Different assessments should be taken into consideration when implementing models on neu-
106 romorphic hardware, especially the trade-offs between simulation time, precision and power consumption.
107 Thus benchmarking neuromorphic hardware with various network models will reveal the advantages and
108 disadvantages of different platforms. In this paper we propose a large dataset of spike-based visual stim-
109uli, NE, and its complementary evaluation methodology. The dataset expands and evolves as research
110 develops and new problems are introduced.

111 In Section 2, some example datasets of conventional non-spiking computer vision are introduced.
112 Section 3 defines the purpose and protocols of the proposed dataset. The sub-datasets and their generation
113 methods are described in detail in Section 4. In accordance with the dataset, its evaluation methodology
114 is demonstrated in Section 5. Moreover, two SNN models are provided as examples of benchmarking
115 hardware platforms in Section 6. Section 7 summarises the paper and discusses future work.

2 RELATED WORK

116 In conventional computer vision, there are a few datasets playing important roles at different times and
117 with various objectives.

2.1 MNIST

118 The MNIST (LeCun et al., 1998) dataset is a subset of the NIST hand written digits dataset. The training
119 set contains 60,000 patterns collected from approximately 250 writers. The testing set is composed of
120 10,000 patterns written by disjoint individuals which were not listed in the training set. All the digits in
121 the dataset are of similar scale centring in a 28×28 image. Due to its straightforward target of classifying

122 real-world images, the plain format of binary data and the simple patterns, MNIST has been one of the
123 most popular datasets in computer vision for over 20 years.

124 Many methods have been verified on this dataset: K-means, SVM, ConvNets, etc. The descending
125 recognition error rate makes it nearly a solved problem, however some modifications, such as position
126 shifts, scaling and noise, bring new challenges. Certainly, a spiking version of the dataset will be an
127 interesting artificial distortion and draw attention to new methods and algorithms on the challenge.

2.2 IMAGENET (Deng et al., 2009)

128 Since the new era of the 4th generation ANN, the DNN, a flow of successful applications have been
129 reported. Meanwhile, training the deeper network triggers a huge demand for sample data. The purpose
130 of putting forward ImageNet was to provide researchers with a large-scale image database, which matches
131 nicely with DNN data requirements. Currently there are 14,197,122 images and 21,841 synsets indexed
132 in the dataset¹. Synsets are meaningful concepts described with a few words or phrases, and they are
133 organised in a hierarchy as in WordNet. The final goal of ImageNet is to provide about 1000 images for
134 each of the 80,000 synsets in WordNet. In other words, there will be tens of millions of images tidily
135 structured, accurately labelled and human annotated. The dataset is a well-recognised benchmark test for
136 the deep learning community, and many attempts have been made to improve the performance of machine
137 learning algorithms on this dataset, for example (Krizhevsky et al., 2012).

2.3 MICROSOFT COCO (Lin et al., 2014)

138 As a good example of a database catching up with state-of-the-art technologies, Microsoft COCO aims to
139 solve three problems in scene understanding by providing large-scale datasets. First is to categorise objects
140 in their non-iconic views, such as being small, ambiguous or partially occluded. Secondly, understanding
141 the context (contextual reasoning) of multiple objects in an image is necessary. Lastly, spatial labelling of
142 the objects is a core analysis in scene understanding. Up to date, the dataset contains 300,000+ images, 2
143 million instances and 5 captions per image.

2.4 ACTION DATASETS

144 Similar examples could be found in video datasets. Two early benchmarks, the KTH (Schilddt et al., 2004)
145 and Weizmann (Blank et al., 2005) datasets, have been used extensively in the past decade. These videos
146 were produced with scripted behaviours in a controlled environment (“in the lab”). They contain single
147 atomic actions, which are simple and neat: walking, running, sitting, etc.

148 Taking the advantages of continuous spiking trains instead of frames of videos, spiking versions of such
149 action datasets will be provided in our future work. A DVS simulation may be needed to convert frames
150 of images into spikes.

151 The YouTube Action Dataset (Liu et al., 2009) targets recognising realistic actions from videos “in the
152 wild”. Thanks to the digital era, unconstrained videos are abundant on the Internet, e.g. YouTube. This
153 YouTube Action dataset is composed of 1,168 videos in 11 categories. The main challenge relies on the
154 massive variations due to the moving camera, background clutter, viewing angles, illuminations and so on.
155 It also aims to detect complex action (non-atomic), e.g. long jump, which consists of several continuous
156 atomic actions.

¹ <http://www.image-net.org/>

3 GUIDING PRINCIPLES

157 The NE database we propose here is a developing and evolving dataset consisting of various spike-based
158 representations of images and videos. The spikes are either generated from spike encoding methods which
159 covert images or frames of videos into spike trains, or recorded from DVS silicon retinas. The spike trains
160 are in the format of AER data, which could easily be used in both event-driven computer simulations and
161 neuromorphic systems. With the NE dataset we hope:

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

4 THE DATASET: NE15-MNIST

180 The name of the first proposed dataset in the benchmarking system is NE15-MNIST which stands for Neu-
181 romorphic Engineering 2015 on MNIST. The original MNIST dataset is downloaded from the website²
182 of THE MNIST DATABASE of handwritten digits (LeCun et al., 1998). The NE15-MNIST is converted
183 into a spiking version of the original dataset consisting of four subsets which were generated for different
184 purposes:

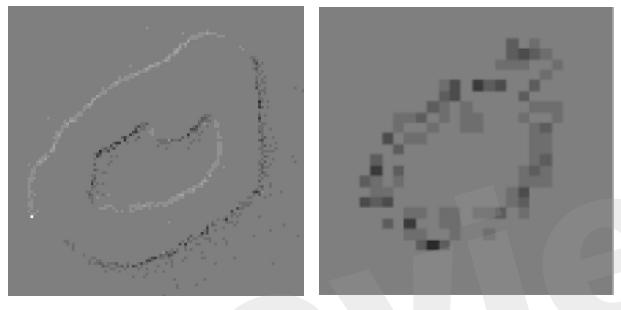
- 185 • *Poissonian* to benchmarking existing methods of rate-based spiking models.
- 186 • *FoCal* to promote the study of spatio-temporal algorithms applied to recognition tasks using few input
187 spikes.
- 188 • *DVS recorded flashing input* to encourage research on fast recognition methods which are found in
189 the primate visual pathway.
- 190 • *DVS recorded moving input* to trigger the study of algorithms targeting on continuous input from
191 real-world sensors and to implement them on mobile neuromorphic robots.

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

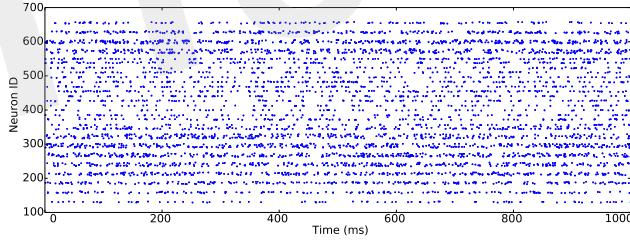
² <http://yann.lecun.com/exdb/mnist/>

4.1 FILE FORMATS

193 Two file formats are supported in the dataset: jAER format (Delbruck, 2008) (.dat or .aedat), and binary
 194 file in NumPy .npy format. The address event representation (AER) interface has been widely used in
 195 neuromorphic systems, especially for vision sensors. The spikes are encoded as time events with corre-
 196 sponding addresses to convey information. The spikes in jAER format, both recorded from a DVS retina
 197 and artificially generated, can be displayed in jAER software. Figure 1a is a snapshot of the software
 198 displaying an .aedat file which is recorded by a DVS retina (Serrano-Gotarredona and Linares-Barranco,
 199 2012). 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.



(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.

4.2 DATA DESCRIPTION

202 4.2.1 *Poissonian* In the cortex, the timing of spikes is highly irregular (Squire and Kosslyn, 1998). It
 203 can be interpreted that the inter-spike interval reflects a random process driven by the instantaneous firing
 204 rate. If the generation of each spike is assumed to be independent of all the other spikes, the spike train
 205 is seen as a Poisson process. The spiking rate can be estimated by averaging the pooled responses of the
 206 neurons.

207 As stated above, rate coding is exclusively used in presenting images with spikes. The spiking rate
 208 of each neuron is in accordance with its corresponding pixel intensity. Instead of providing exact spike
 209 arrays, we share the Python code for generating the spikes. Every recognition system may require different
 210 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 Fig. 1a is converted to Poissonian spike trains, see Fig. 1b. The raster plot can be found in Fig. 1c, indicating the intensities of the pixels.

4.2.2 *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 Filter overlap Correction algorithm or FoCal (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).

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

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 (Fig. 2) is inversely proportional to a spike emission time (i.e. the higher the pixel value, the sooner the spike will be sent out.)

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).

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 Fig. 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.

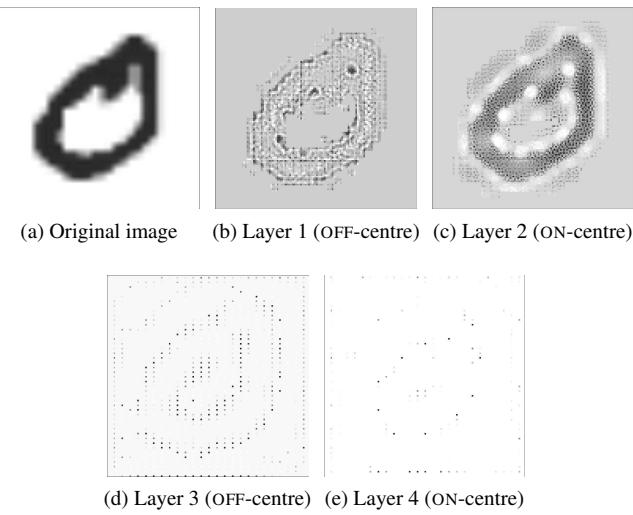


Figure 2: Results of correcting the spikes from the simulated ganglion cell layers using the FoCal algorithms.

Algorithm 1 FoCal, Part 2

```

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

```

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

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

247 4.2.3 *DVS Sensor Output with Flashing Input* The purpose of including the subset of DVS recorded
248 flashing digits is to promote the application of Rank-Order-Coding to DVS output, and accelerate the fast
249 on-set recognition by using just the beginning part of spike trains within less than 30 ms.

250 Each digit and a blank image was shown alternately and each display lasted one second. The digits were
251 displayed on an LCD monitor in front of the DVS retina ([Serrano-Gotarredona and Linares-Barranco](#),

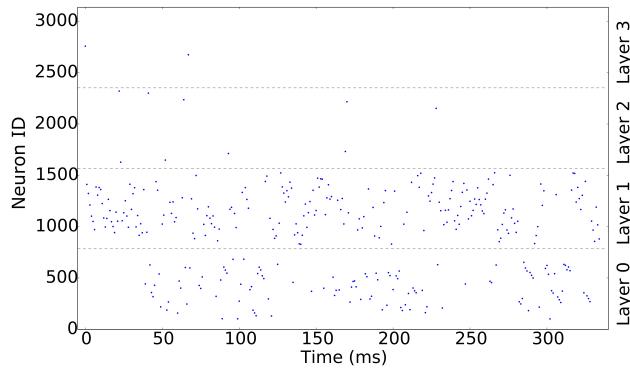


Figure 3: First 30% of the rank-order encoded spikes produced with FoCal.

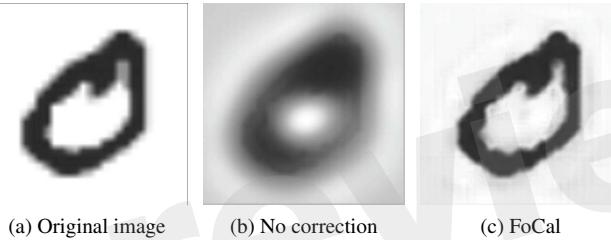


Figure 4: Reconstruction result comparison.

252 2013) and were placed in the centre of the visual field of the camera. Since there are two polarities of the
 253 spikes: 'ON' indicates the increase of the intensity while 'OFF' reflects the opposite, there are 'ON' and
 254 'OFF' flashing recordings respectively per digit. In Fig. 5, the burstiness of the spikes is illustrated where
 255 most of the spikes occur in a 30 ms slot. In total, the subset of the database contains $2 \times 60,000$ recordings
 256 for training and $2 \times 10,000$ for testing.

257 4.2.4 *DVS Sensor Output with Moving Input* In order to address the problems of position- and scale-
 258 invariance, a subset of DVS recorded moving digits is presented.

259 MNIST digits were scaled to three different sizes, by using smooth interpolation algorithms to increase
 260 their size from the original 28x28 pixel size, and displayed on the monitor with slow motion. The same
 261 DVS (Serrano-Gotarredona and Linares-Barranco, 2013) used in Section 4.2.3 captured the movements of
 262 the digits and generated spike trains for each pixel of its 128×128 resolution. A total of 30,000 recordings
 263 were made: 10 digits, at 3 different scales, 1000 different handwritten samples for each.

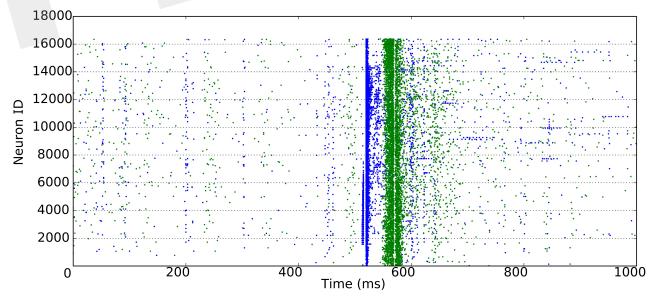
5 PERFORMANCE EVALUATION

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

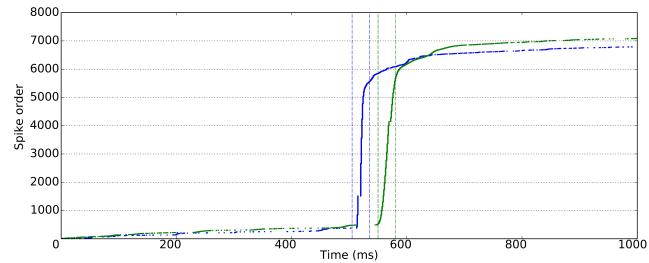
5.1 HARDWARE-INDEPENDENT

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, 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



(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 appear 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 FCnet, LIF neurons | Off-line trained with ReLU, weight normalization | 99.1% (ConvNet), 98.6% (FCnet); 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 | FC decision layer, LIF neurons | K-means clusters, Supervised STDP 18,000 s of training | 92.98% 1 s per test 10.70 ms latency |

289 the response time (latency) of an SNN model. A faster model is more suitable for real-time recognition
 290 systems such as neuromorphic robotics. A commonly reported characteristic is the accuracy of the
 291 network, perhaps adding remarks on how these scores are obtained could help to unify criteria and ease
 292 comparison. Work on SNN-based classifications of MNIST are listed in Table 2 and evaluated on the
 293 proposed metrics.

5.2 HARDWARE-SPECIFIC

294 Depending on how neurons, synapses and spike transmission are implemented neuromorphic systems
 295 can be categorised as either analogue, digital, or mixed-mode analogue/digital VLSI circuits. Some
 296 analogue implementations exploit sub-threshold transistor dynamics to emulate neurons and synapses

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 (4 signed int + on/off state) | 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 | 198 pJ/SE 13.5 MSops/W (network only) |
| iAER-IFAT (Yu et al., 2012) | Mixed-mode, Scalable | Fixed models, Config params | No plasticity | Analogue neuron/synapse | Real-time | 20GSops/W |

297 directly on hardware (Indiveri et al., 2011) and are more energy-efficient while requiring less area than
 298 their digital counterparts (Joubert et al., 2012). However, the behaviour of analogue circuits is largely
 299 determined during the fabrication process due to transistor mismatch (Indiveri et al., 2011; Pedram and
 300 Nazarian, 2006; Linares-Barranco et al., 2003), while their wiring densities render them impractical for
 301 large-scale systems. The majority of mixed-mode analogue/digital neuromorphic platforms, such as the
 302 High Input Count Analog Neural Network (HI-CANN) (Schemmel et al., 2010), Neurogrid (Benja-
 303 min et al., 2014), HiAER-IFAT (Yu et al., 2012), use analogue circuits to emulate neurons and digital
 304 packet-based technology to communicate spikes as AER events. This enables reconfigurable connectivity
 305 patterns, while the time of spikes is expressed implicitly since typically a spike reaches its destination
 306 in less than a millisecond, thus fulfilling the real-time requirement. Digital neuromorphic platforms such
 307 as TrueNorth (Merolla et al., 2014) use digital circuits with finite precision to simulate neurons in an
 308 event driven manner to minimise the active power dissipation. Neuromorphic systems suffer from model
 309 flexibility, since neurons and synapses are fabricated directly on hardware with only a small subset of
 310 parameters exposed to the researcher. SpiNNaker is a biologically inspired, massively-parallel, scalable
 311 computing architecture designed by the Advanced Processor Technologies (APT) group at the University
 312 of Manchester. SpiNNaker has been optimised to simulate very large-scale spiking neural networks

313 in real-time (Furber et al., 2014). SpiNNaker aims to combine the advantages of conventional computers
 314 and neuromorphic hardware by utilising low-power programmable cores and scalable event-driven
 315 communications hardware.

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

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

6 CASE STUDIES

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

6.1 CASE STUDY I

349 The first case study is a simple two-layered network where the input neurons receive Poissonian presented
 350 spike trains from the dataset and form an FC network with the decision neurons. The model utilises LIF
 351 neurons, and the parameters are all with biological means, see the listed values in Table 4. The LIF neuron
 352 model follows the membrane potential dynamics:

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

353 where τ_m is the membrane time constant, V_{rest} is the resting potential, R_m is the membrane resistance and
 354 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
 355 capacitance. A spike is generated when the membrane potential goes beyond the threshold, V_{thresh} and the
 356 membrane potential resets to V_{reset} . In addition, a neuron cannot fire within the refractory period, τ_{refrac} ,
 357 after generating a spike.

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

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

368 In order to fully assess the performance, different settings have been configured on the network, such as
 369 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 |

370

371 **6.1.1 Training** There are two layers in the model: 28×28 input neurons fully connect to 100 decision
 372 neurons. Each decision neuron responds to a certain template of a digit. In the standard configuration, there
 373 are 10 decision neurons answering to the same digit with slightly different templates. Those templates are
 374 embedded in the connection weights between the two layers. Fig. 6a shows how the connections to a
 375 single decision neuron are tuned.

376 The training set of 60,000 hand written digits are firstly classified into 100 classes, 10 subclasses per
 377 digit, using K-means clusters. So the images in a certain subclass are used to train one corresponding
 378 decision neuron. The firing rates of the input neurons are assigned linearly according to their intensities
 379 and normalised with a total firing rate of 2,000 Hz. All the images together are presented for 18,000 s
 380 (about 300 ms per image) during training and at the same time a teaching signal of 50 Hz is conveyed
 381 to the decision neuron to trigger STDP learning. The trained weights are plotted in accordance with the
 382 positions of the decision neurons in Fig. 6b.

383 **6.1.2 Testing** After training the weights of the plastic synapses are set to static, keeping the state of
 384 the weights at the last moment of training. The weak weights were set to inhibitory connections with an
 385 identical strength. The feed-forward testing network is shown in Fig. 6b where Poissonian spike trains
 386 are generated the same way as in the training with a total firing rate of 2,000 Hz per image. The input
 387 neurons convey the same spike trains to every decision neuron through its responding trained synaptic

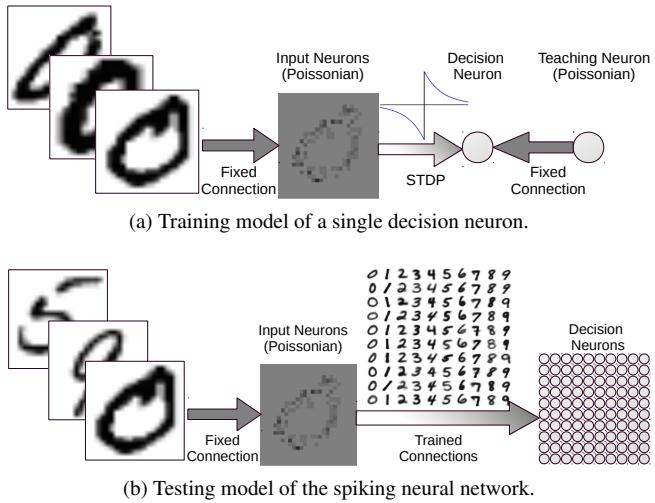


Figure 6: The training and testing model of the two-layered spiking neural network.

388 weights. Every testing image (10,000 images in total) is presented once and lasts 1 s with a silence of
 389 200 ms between them. The output neuron with the highest firing rate decides what digit was recognized.
 390 Taken the trained weights from the NEST simulation, the accuracy of the recognition on NEST reaches
 391 90.03% with the standard configuration, while the result drops slightly to 89.97% using SpiNNaker. In
 392 comparison, both trained and tested on SpiNNaker the recognition accuracy is 87.41%, and with the same
 393 weights applied to NEST the result turns out to be 87.25%.

394 **6.1.3 Evaluation** The evaluation starts from the hardware-independent side, focusing on the spike-
 395 based recognition analysis. As mentioned in Section 5.1, CA and response time (latency) are the main
 396 concerns when assessing the recognition capability. In our experiment, two sets of weights were applied:
 397 the original STDP trained weights and scaled-up weights which are 10 times stronger. The spiking rates
 398 of the testing samples were also modified, ranging from 10 to 5,000 Hz.

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

408 The latency of an SNN model is the result of the input rates and synaptic weights. As the input rates
 409 grow, there are more spikes arriving at the decision neurons, triggering them to spike sooner. A similar idea
 410 applies to the influence of synaptic weights. If stronger weights are taken, then the membrane potential
 411 of a neuron reaches its threshold earlier. Fig. 7d indicates that the latency is shortened with increasing
 412 input rates with both the original and scaled-up weights. When the spiking rate is less than 2,000 Hz, the
 413 network with stronger weights has a much shorter latency. As long as there are enough spikes to trigger
 414 the decision neurons to spike, increasing the test time will not make the network respond sooner (Fig. 7c.)
 415

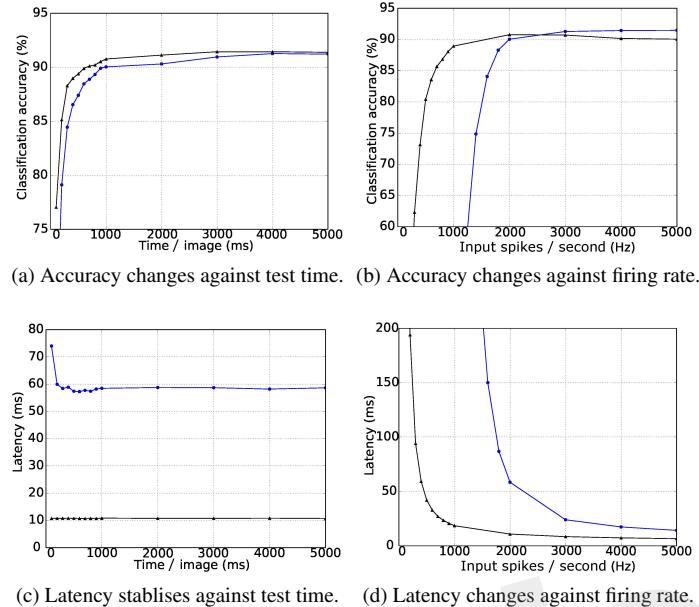


Figure 7: Accuracy and response time (latency) change over test time and input firing rate per testing image. Original trained weights are used (circles in blue) as well as the scaled up ($\times 10$) weights (triangles in black).

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. 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 W³. 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 Fig. 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.

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

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

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

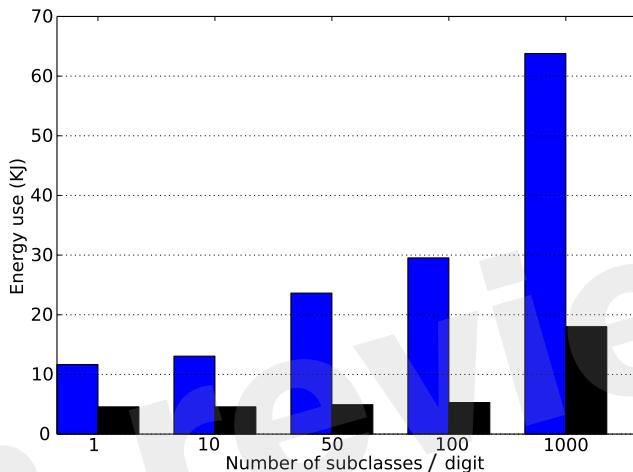


Figure 8: Energy usages of different network size both using NEST (blue) on a PC and SpiNNaker (black).

6.2 CASE STUDY II

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

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

446 O’Connor et al. (2013) proposed a method to map off-line trained DBNs into a spiking neural networks
 447 and take advantage of the real-time performance and energy efficiency of neuromorphic platforms. This
 448 led initially to an implementation on an event-driven Field-Programmable Gate Array (FPGA) called
 449 Minitaur (Neil and Liu, 2014) and then on the SpiNNaker platform (Stromatias et al., 2015a). For this
 450 work we used an off-line trained⁴ spiking DBN with a 784-500-500-10 network topology. Simulations
 451 take place on a software spiking neural network simulator named Brian (Goodman and Brette, 2008) and
 452 results are verified on the SpiNNaker platform.

⁴ <https://github.com/dannyneil/edbn/>

453 6.2.1 *Training* DBNs consist of stacked Restricted Boltzmann Machines (RBMs), which are fully
 454 connected recurrent networks but without any connections between neurons of the same layer. Training
 455 is performed unsupervised using the standard CD rule (Hinton et al., 2006) and only the output layer
 456 is trained in a supervised manner. The main difference between spiking DBNs and traditional DBNs
 457 is the activation function used for the neurons. O'Connor et al. (2013) proposed the use of the Siegert
 458 approximation (Jug et al., 2012) as the activation function, which returns the expected firing rate of a LIF
 459 neuron given input firing rates, input weights, and standard neuron parameters.

460 6.2.2 *Testing* After the training process the learnt synaptic weights can be used in a spiking neural
 461 network which consists of LIF neurons with delta-current synapses. Table 6 shows the LIF parameters
 462 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 |

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

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

468 6.2.3 *Evaluation* Neuromorphic platforms may have limited hardware resources to store the synaptic
 469 weights (Schemmel et al., 2010; Merolla et al., 2014). In order to investigate how the precision of the
 470 weights affects the CA of a spiking DBN the double floating point weights of the offline trained network
 471 were converted to different fixed-point representations. The following notation will be used throughout
 472 this paper, $Qm.f$, where m signifies the number of bits for the integer part (including the sign bit) and f the
 473 number of bits used for the fractional part.

474 Figure 9 shows the effect of reduced weight bit precision on the CA for different input firing rates on the
 475 Brian simulator. Using the same weight precision of Q3.8, SpiNNaker achieved a CA of 94.94% when
 476 1,500 Hz was used for the input population (Stromatias et al., 2015a). Brian for the same firing rates
 477 and weight precision achieved a CA of 94.955%. Results are summarised in Table 7. The slightly lower
 478 CA of the SpiNNaker simulation indicates that not only the weight precision but also the precision of the
 479 membrane potential affects the overall classification performance.

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 |

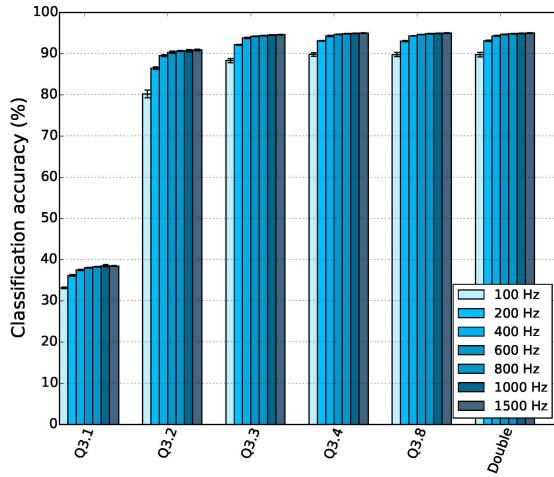


Figure 9: CA as a function of the weight bit precision for different input firing rates.

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

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

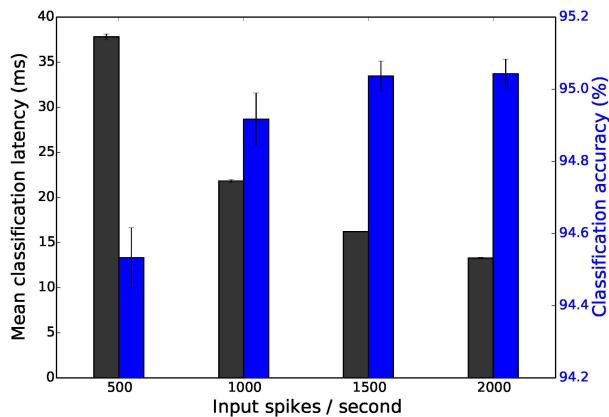


Figure 10: Mean classification latency (black) and classification accuracy (blue) as a function of the input spikes per second for the spiking DBN. Results are averaged over 4 trials, error bars show standard deviations.

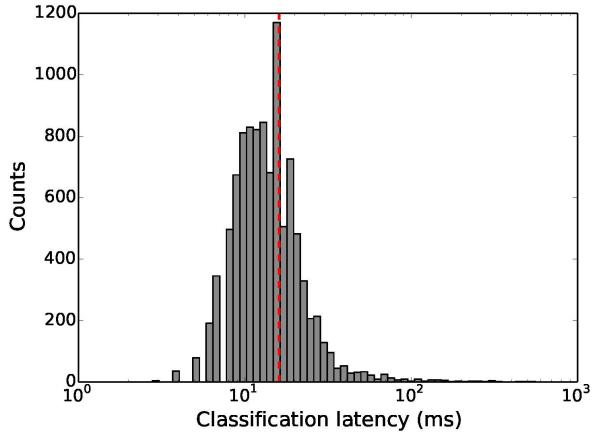


Figure 11: Histogram of the classification latencies for the MNIST digits of the testing set when the input rates are set to 1,500 Hz. The mean classification latency of the spiking DBN on SpiNNaker is 16 ms.

490 Finally, this particular spiking DBN ran on a single SpiNNaker chip (16 ARM9 cores) and dissipated
 491 less than 0.3 W when 2,000 spikes per second per digit were used, as seen in Figure 12. The identical
 492 network ran on Minitaur (Neil and Liu, 2014), an event-driven FPGA implementation, and consumed
 493 1.5 W when 1,000 spikes per image were used.

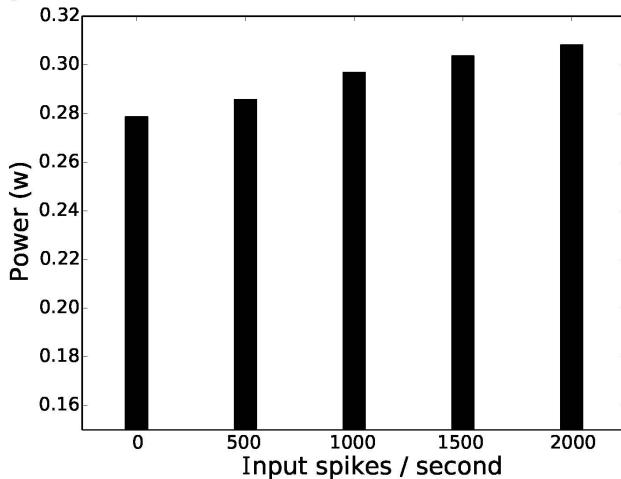


Figure 12: Power dissipation of a spiking DBN running on a single SpiNNaker chip as a function of the total number of input spikes per second.

7 CONCLUSION

7.1 WHAT WE HAVE SAID AND DONE

494 This paper puts forward the NE dataset as a baseline for comparisons on vision based SNNs. It contains
495 converted spike representations of existing widely-used databases in the vision recognition field. Since
496 new problems will be introduced continuously before vision becomes a solved question, the dataset will
497 evolve as research develops. The conversion methods transforming images and videos to spike trains
498 will advance. The number of vision databases included will increase and the corresponding evaluation
499 methodology will evolve as well. The dataset aims to provide a unified spike-based vision database and
500 complementary evaluation methodologies to assess the performance of various SNN algorithms.

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

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

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

525 Although spike-based algorithms have not surpassed their non-spiking counterparts in terms of recogni-
526 tion accuracy, they have shown great performance in response time and energy efficiency. The dataset
527 makes the comparison of SNNs with conventional recognition methods possible by using converted spike
528 presentations of the same vision databases. As the dataset grows, it will allow new problems to be inves-
529 tigated by researchers, which should allow the identification of future directions and, in consequence,
530 advance the field.

7.2 THE FUTURE DIRECTION OF AN EVOLVING DATABASE

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

538 our plan is also to embed modulated trajectory information to direct the recording using DVS sensors to
539 simulate human saccades.

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

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

ACKNOWLEDGMENTS

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

REFERENCES

- 561 Benjamin, B. V., Gao, P., McQuinn, E., Choudhary, S., Chandrasekaran, A. R., Bussat, J.-M., et al. (2014).
562 Neurogrid: a mixed-analog-digital multichip system for large-scale neural simulations. *Proceedings of
563 the IEEE* 102, 699–716
- 564 Beyeler, M., Dutt, N. D., and Krichmar, J. L. (2013). Categorization and decision-making in a neu-
565 robiologically plausible spiking network using a STDP-like learning rule. *Neural Networks* 48,
566 109–124
- 567 Bichler, O., Querlioz, D., Thorpe, S. J., Bourgoin, J.-P., and Gamrat, C. (2012). Extraction of tempo-
568 rally correlated features from dynamic vision sensors with spike-timing-dependent plasticity. *Neural
569 Networks* 32, 339–348
- 570 Bill, J. and Legenstein, R. (2014). A compound memristive synapse model for statistical learning through
571 STDP in spiking neural networks. *Frontiers in neuroscience* 8
- 572 Blank, M., Gorelick, L., Shechtman, E., Irani, M., and Basri, R. (2005). Actions as space-time shapes. In
573 *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*. vol. 2, 1395–1402
- 574 Brader, J. M., Senn, W., and Fusi, S. (2007). Learning real-world stimuli in a neural network with
575 spike-driven synaptic dynamics. *Neural computation* 19, 2881–2912
- 576 Camunas-Mesa, L., Zamarreño-Ramos, C., Linares-Barranco, A., Acosta-Jiménez, A. J., Serrano-
577 Gotarredona, T., and Linares-Barranco, B. (2012). An event-driven multi-kernel convolution processor
578 module for event-driven vision sensors. *Solid-State Circuits, IEEE Journal of* 47, 504–517
- 579 Cao, Y., Chen, Y., and Khosla, D. (2015). Spiking deep convolutional neural networks for energy-efficient
580 object recognition. *International Journal of Computer Vision* 113, 54–66

- 581 Davison, A. P., Brüderle, D., Eppler, J., Kremkow, J., Muller, E., Pecevski, D., et al. (2008). PyNN: a
582 common interface for neuronal network simulators. *Frontiers in neuroinformatics* 2
- 583 Delbrück, T. (2008). Frame-free dynamic digital vision. In *Proceedings of Intl. Symp. on Secure-Life*
584 *Electronics, Advanced Electronics for Quality Life and Society*. 21–26
- 585 Delorme, A. and Thorpe, S. J. (2001). Face identification using one spike per neuron: resistance to image
586 degradations. *Neural Networks* 14, 795–803
- 587 Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: a large-scale hierarchi-
588 cal image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference*
589 *on*. 248–255
- 590 DiCarlo, J. J., Zoccolan, D., and Rust, N. C. (2012). How does the brain solve visual object recognition?
591 *Neuron* 73, 415–434
- 592 Diehl, P., Neil, D., Binas, J., Cook, M., Liu, S.-C., and Pfeiffer, M. (2015). Fast-classifying, high-accuracy
593 spiking deep networks through weight and threshold balancing. In *Neural Networks (IJCNN), The 2015*
594 *International Joint Conference on*. to be published
- 595 Diehl, P. U. and Cook, M. (2015). Unsupervised learning of digit recognition using spike-timing-
596 dependent plasticity. *Frontiers in Computational Neuroscience* 9, 99
- 597 Eliasmith, C., Stewart, T. C., Choo, X., Bekolay, T., DeWolf, T., Tang, Y., et al. (2012). A large-scale
598 model of the functioning brain. *science* 338, 1202–1205
- 599 Felleman, D. J. and Van Essen, D. C. (1991). Distributed hierarchical processing in the primate cerebral
600 cortex. *Cerebral cortex* 1, 1–47
- 601 Fu, S.-Y., Yang, G.-S., and Kuai, X.-K. (2012). A spiking neural network based cortex-like mechanism
602 and application to facial expression recognition. *Computational intelligence and neuroscience* 2012, 19
- 603 Furber, S. and Temple, S. (2007). Neural systems engineering. *Journal of the Royal Society interface* 4,
604 193–206
- 605 Furber, S. B., Galluppi, F., Temple, S., Plana, L., et al. (2014). The SpiNNaker Project. *Proceedings of*
606 *the IEEE* 102, 652–665
- 607 Gewaltig, M.-O. and Diesmann, M. (2007). NEST (NEural Simulation Tool). *Scholarpedia* 2, 1430
- 608 Goodman, D. and Brette, R. (2008). Brian: a simulator for spiking neural networks in Python. *Frontiers*
609 *in neuroinformatics* 2
- 610 Hinton, G. E., Osindero, S., and Teh, Y.-W. (2006). A fast learning algorithm for Deep Belief Nets.
611 *Neural computation* 18, 1527–1554
- 612 Hof, R. (2013). 10 breakthrough technologies 2013
- 613 Hopkins, M. and Furber, S. (2015 to be published). Accuracy and Efficiency in Fixed-Point Neural ODE
614 Solvers. *Neural computation*
- 615 Indiveri, G., Linares-Barranco, B., Hamilton, T. J., Van Schaik, A., Etienne-Cummings, R., Delbrück, T.,
616 et al. (2011). Neuromorphic silicon neuron circuits. *Frontiers in neuroscience* 5
- 617 Joubert, A., Belhadj, B., Temam, O., and Héliot, R. (2012). Hardware spiking neurons design: analog or
618 digital? In *Neural Networks (IJCNN), The 2012 International Joint Conference on* (IEEE), 1–5
- 619 Jug, F., Lengler, J., Krautz, C., and Steger, A. (2012). Spiking networks and their rate-based equivalents:
620 does it make sense to use Siegert neurons? In *Swiss Soc. for Neuroscience*
- 621 Kolb, H. (2003). How the retina works. *American scientist* 91, 28–35
- 622 Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet classification with deep convolutional
623 neural networks. In *Advances in neural information processing systems*. 1097–1105
- 624 Lagorce, X., Stromatias, E., Galluppi, F., Plana, L. A., Liu, S.-C., Furber, S. B., et al. (2015). Breaking
625 the millisecond barrier on SpiNNaker: implementing asynchronous event-based plastic models with
626 microsecond resolution. *Frontiers in Neuroscience* 9, 206
- 627 LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document
628 recognition. *Proceedings of the IEEE* 86, 2278–2324
- 629 LeCun, Y., Huang, F. J., and Bottou, L. (2004). Learning methods for generic object recognition with
630 invariance to pose and lighting. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004.*
631 *Proceedings of the 2004 IEEE Computer Society Conference on*. vol. 2, II–97
- 632

- 633 Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., et al. (2014). Microsoft COCO:
634 Common Objects in COnText. In *Computer Vision–ECCV 2014* (Springer). 740–755
- 635 Linares-Barranco, B., Serrano-Gotarredona, T., and Serrano-Gotarredona, R. (2003). Compact low-
636 power calibration mini-DACs for neural arrays with programmable weights. *Neural Networks, IEEE
637 Transactions on* 14, 1207–1216
- 638 Liu, J., Luo, J., and Shah, M. (2009). Recognizing realistic actions from videos “in the wild”. In *Computer
639 Vision and Pattern Recognition, 2009. CVPR. IEEE Conference on.* 1996–2003
- 640 Liu, Q. and Furber, S. (2015). Real-time recognition of dynamic hand postures on a neuromorphic system.
641 In *Artificial Neural Networks, 2015. ICANN. International Conference on.* vol. 1, 979
- 642 Lyons, M., Akamatsu, S., Kamachi, M., and Gyoba, J. (1998). Coding facial expressions with gabor
643 wavelets. In *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International
644 Conference on.* 200–205
- 645 Masmoudi, K., Antonini, M., Kornprobst, P., and Perrinet, L. (2010). A novel bio-inspired static image
646 compression scheme for noisy data transmission over low-bandwidth channels. In *Acoustics Speech
647 and Signal Processing (ICASSP), 2010 IEEE International Conference on.* 3506–3509
- 648 Matsugu, M., Mori, K., Ishii, M., and Mitarai, Y. (2002). Convolutional spiking neural network model
649 for robust face detection. In *Neural Information Processing, 2002. ICONIP'02. Proceedings of the 9th
650 International Conference on.* vol. 2, 660–664
- 651 Merolla, P. A., Arthur, J. V., Alvarez-Icaza, R., Cassidy, A. S., Sawada, J., Akopyan, F., et al. (2014). A
652 million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*
653 345, 668–673
- 654 Neftci, E., Das, S., Pedroni, B., Kreutz-Delgado, K., and Cauwenberghs, G. (2013). Event-driven
655 contrastive divergence for spiking neuromorphic systems. *Frontiers in neuroscience* 7
- 656 Neil, D. and Liu, S.-C. (2014). Minitaur, an event-driven FPGA-based spiking network accelerator. *Very
657 Large Scale Integration (VLSI) Systems, IEEE Transactions on* 22, 2621–2628
- 658 Nessler, B., Pfeiffer, M., Buesing, L., and Maass, W. (2013). Bayesian computation emerges in generic
659 cortical microcircuits through spike-timing-dependent plasticity. *PLoS Comput Biol*
- 660 O'Connor, P., Neil, D., Liu, S.-C., Delbruck, T., and Pfeiffer, M. (2013). Real-time classification and
661 sensor fusion with a spiking deep belief network. *Frontiers in neuroscience* 7
- 662 Pedram, M. and Nazarian, S. (2006). Thermal modeling, analysis, and management in VLSI circuits:
663 principles and methods. *Proceedings of the IEEE* 94, 1487–1501
- 664 Posch, C., Serrano-Gotarredona, T., Linares-Barranco, B., and Delbruck, T. (2014). Retinomorphic
665 event-based vision sensors: bioinspired cameras with spiking output. *Proceedings of the IEEE* 102,
666 1470–1484
- 667 Riesenhuber, M. and Poggio, T. (1999). Hierarchical models of object recognition in cortex. *Nature
668 neuroscience* 2, 1019–1025
- 669 Schemmel, J., Bruderle, D., Grubl, A., Hock, M., Meier, K., and Millner, S. (2010). A wafer-scale
670 neuromorphic hardware system for large-scale neural modeling. In *Circuits and Systems (ISCAS),
671 Proceedings of 2010 IEEE International Symposium on.* 1947–1950
- 672 Schüldt, C., Laptev, I., and Caputo, B. (2004). Recognizing human actions: a local SVM approach. In
673 *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on* (IEEE),
674 vol. 3, 32–36
- 675 Sen, B. and Furber, S. (2009). Evaluating rank-order code performance using a biologically-derived retinal
676 model. In *Neural Networks, 2009. IJCNN. International Joint Conference on* (IEEE), 2867–2874
- 677 Serrano-Gotarredona, T. and Linares-Barranco, B. (2013). A 128×128 1.5% contrast sensitivity
678 0.9% FPN $3\mu s$ latency 4 mW asynchronous frame-free dynamic vision sensor using transimpedance
679 preamplifiers. *Solid-State Circuits, IEEE Journal of* 48, 827–838
- 680 Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., and Poggio, T. (2007). Robust object recognition
681 with cortex-like mechanisms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 29,
682 411–426
- 683 Sharp, T., Galluppi, F., Rast, A., and Furber, S. (2012). Power-efficient simulation of detailed cortical
684 microcircuits on SpiNNaker. *Journal of neuroscience methods* 210, 110–118

- 685 Squire, L. R. and Kosslyn, S. M. (1998). *Findings and current opinion in cognitive neuroscience* (MIT
686 Press)
- 687 Stromatias, E., Galluppi, F., Patterson, C., and Furber, S. (2013). Power analysis of large-scale, real-time
688 neural networks on SpiNNaker. In *Neural Networks (IJCNN), The 2013 International Joint Conference*
689 *on*. 1–8
- 690 Stromatias, E., Neil, D., Galluppi, F., Pfeiffer, M., Liu, S.-C., and Furber, S. (2015a). Scalable energy-
691 efficient, low-latency implementations of trained spiking deep belief networks on SpiNNaker. In *Neural*
692 *Networks (IJCNN), The 2015 International Joint Conference on* (IEEE), to be published
- 693 Stromatias, E., Neil, D., Pfeiffer, M., Galluppi, F., Furber, S. B., and Liu, S.-C. (2015b). Robustness of
694 spiking deep belief networks to noise and reduced bit precision of neuro-inspired hardware platforms.
695 *Frontiers in neuroscience* 9
- 696 Van Rullen, R. and Thorpe, S. J. (2001). Rate coding versus temporal order coding: what the retinal
697 ganglion cells tell the visual cortex. *Neural computation* 13, 1255–1283
- 698 Van Rullen, R. and Thorpe, S. J. (2002). Surfing a spike wave down the ventral stream. *Vision research*
699 42, 2593–2615
- 700 Yang, M., Liu, S.-C., and Delbrück, T. (2015). A dynamic vision sensor with 1% temporal contrast
701 sensitivity and in-pixel asynchronous delta modulator for event encoding. *Solid-State Circuits, IEEE*
702 *Journal of* 50, 2149–2160
- 703 Yarbus, A. L. (1967). *Eye movements during perception of complex objects* (Springer)
- 704 Yu, T., Park, J., Joshi, S., Maier, C., and Cauwenberghs, G. (2012). 65k-neuron Integrate-and-Fire array
705 transceiver with address-event reconfigurable synaptic routing. In *Biomedical Circuits and Systems*
706 *Conference (BioCAS), 2012 IEEE*. 21–24
- 707 Zhao, B., Ding, R., Chen, S., Linares-Barranco, B., and Tang, H. (2015). Feedforward categorization
708 on AER motion events using cortex-like features in a spiking neural network. *Neural Networks and*
709 *Learning Systems, IEEE Transactions on* 26, 1963–1978

Figure 1.JPEG

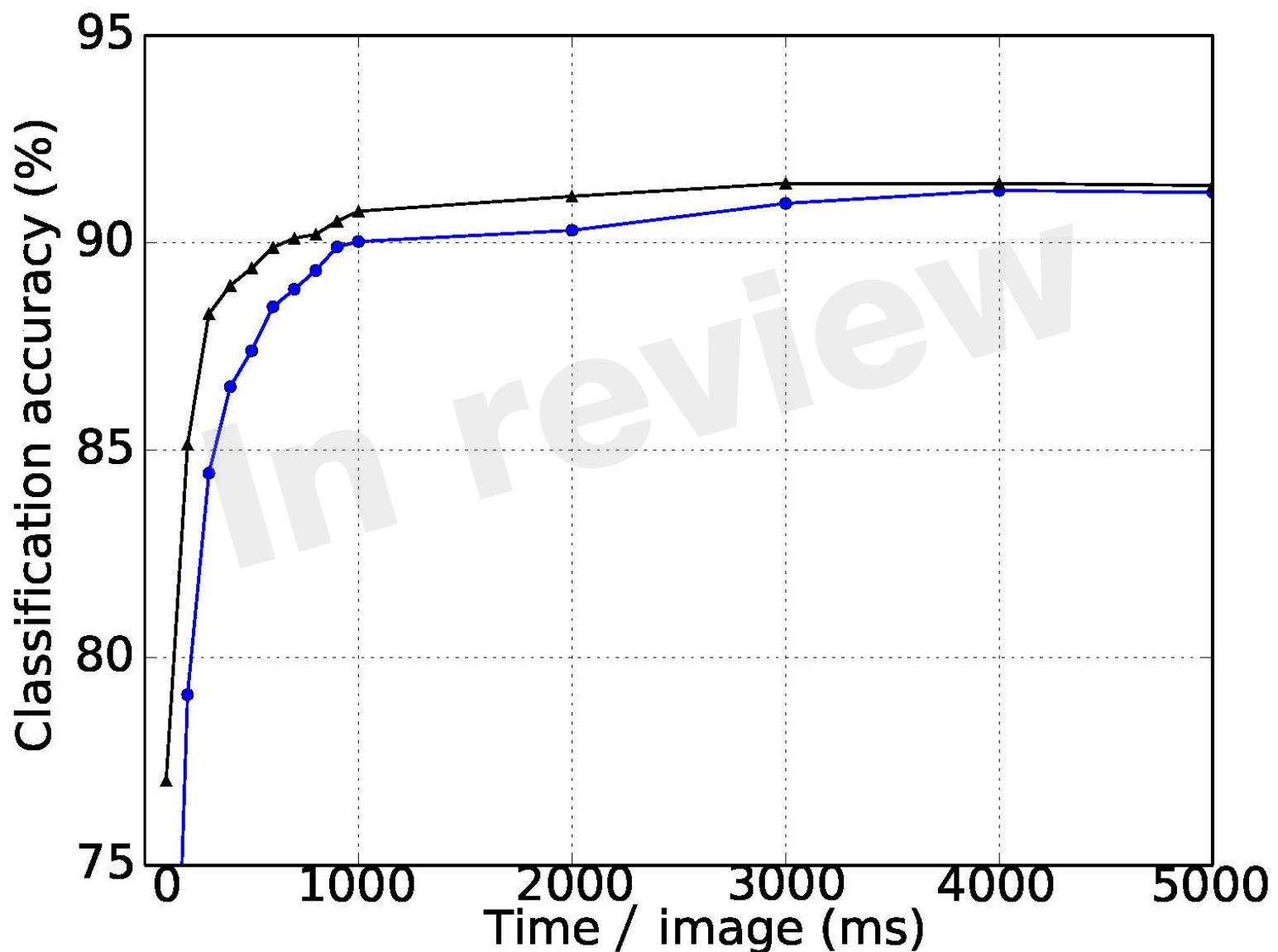


Figure 2.JPEG

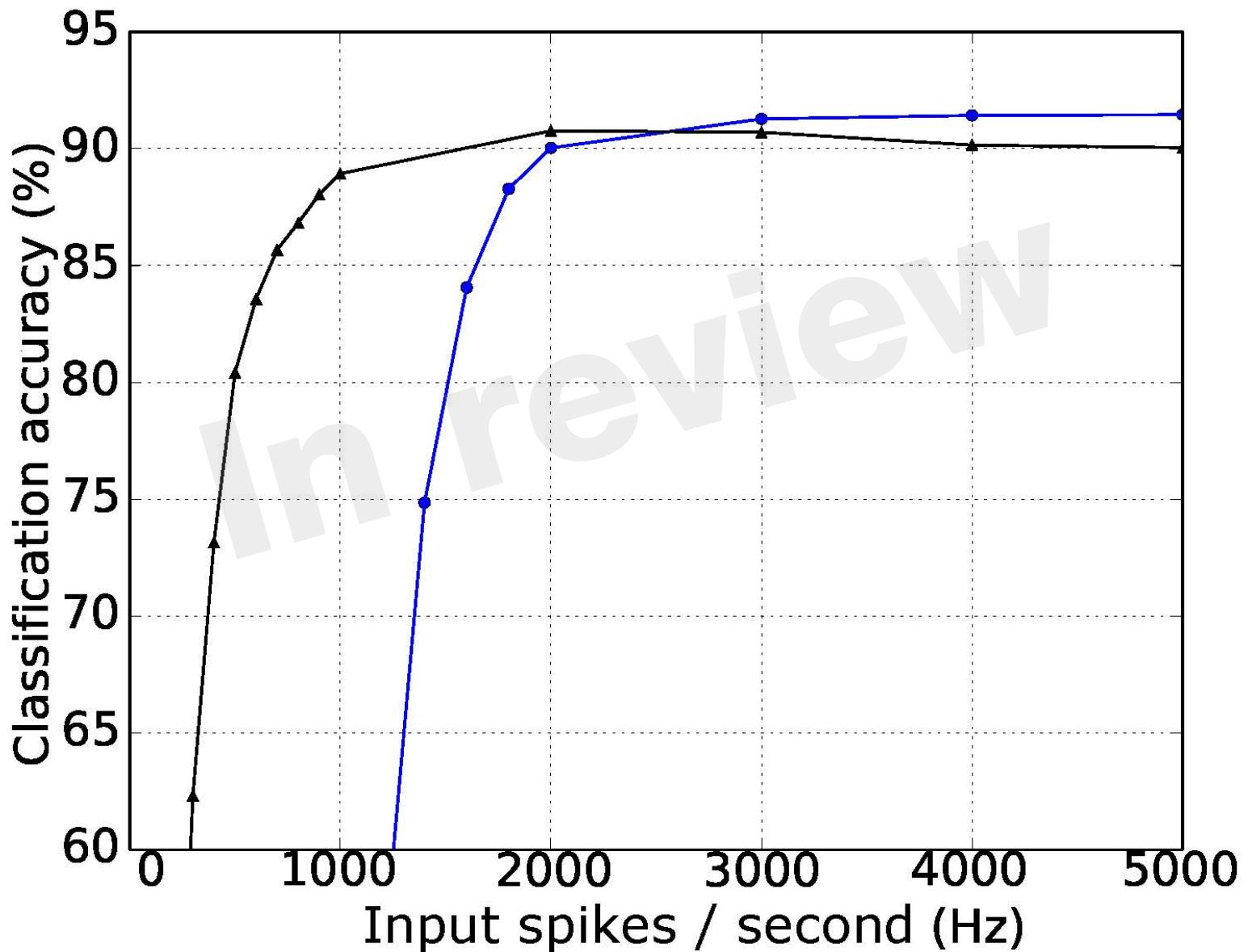


Figure 3.JPEG

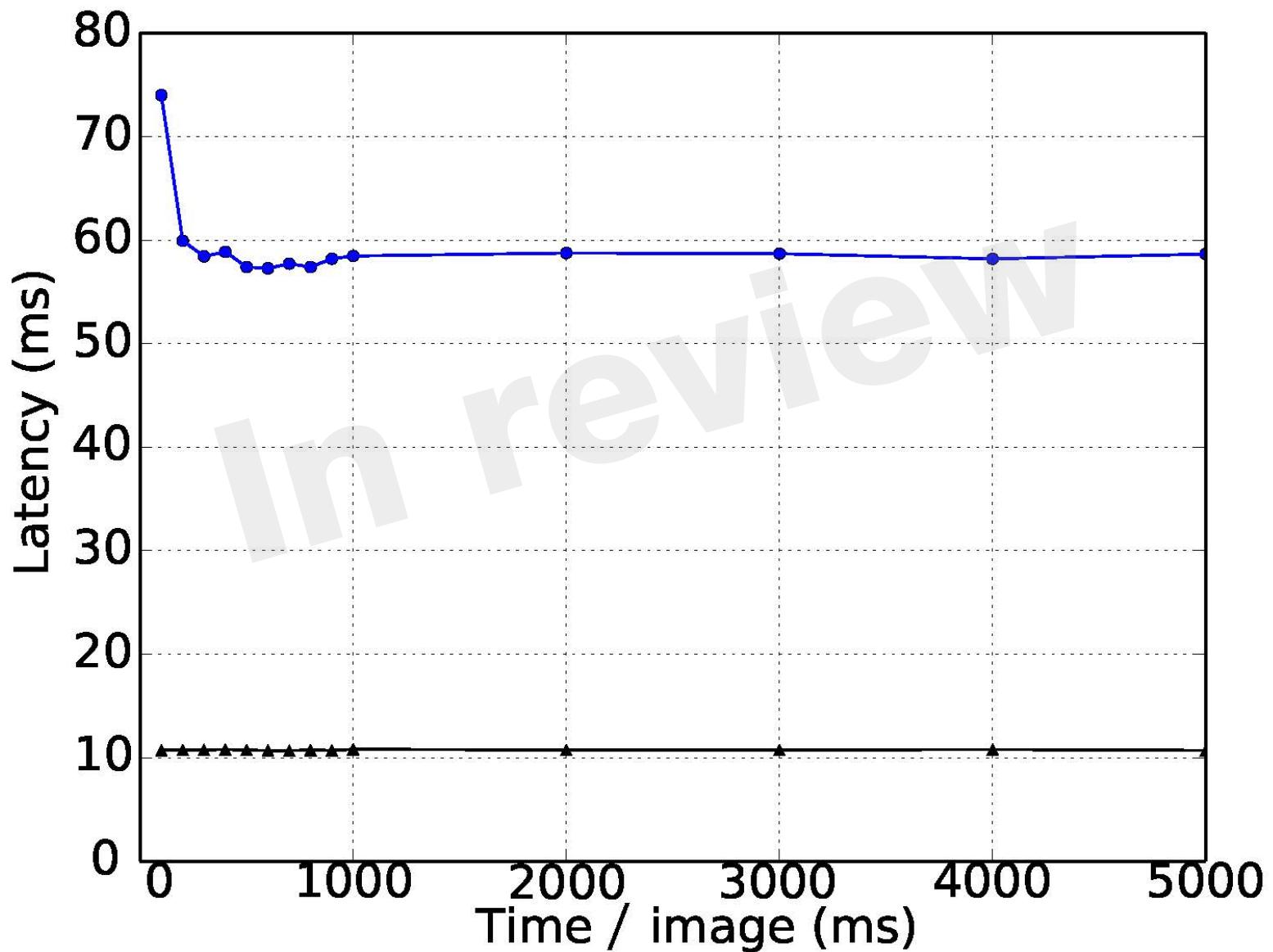


Figure 4.JPEG

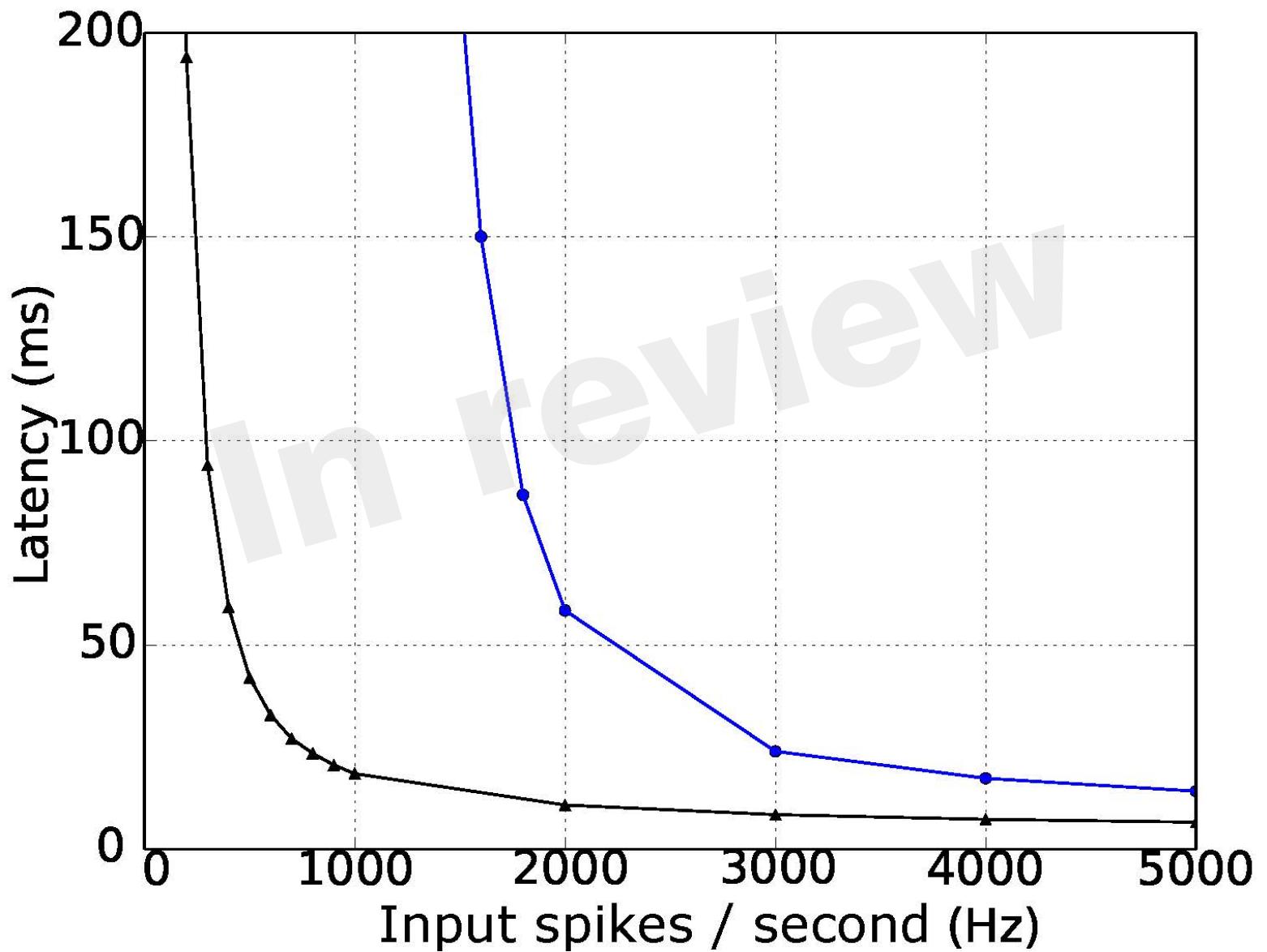


Figure 5.JPEG

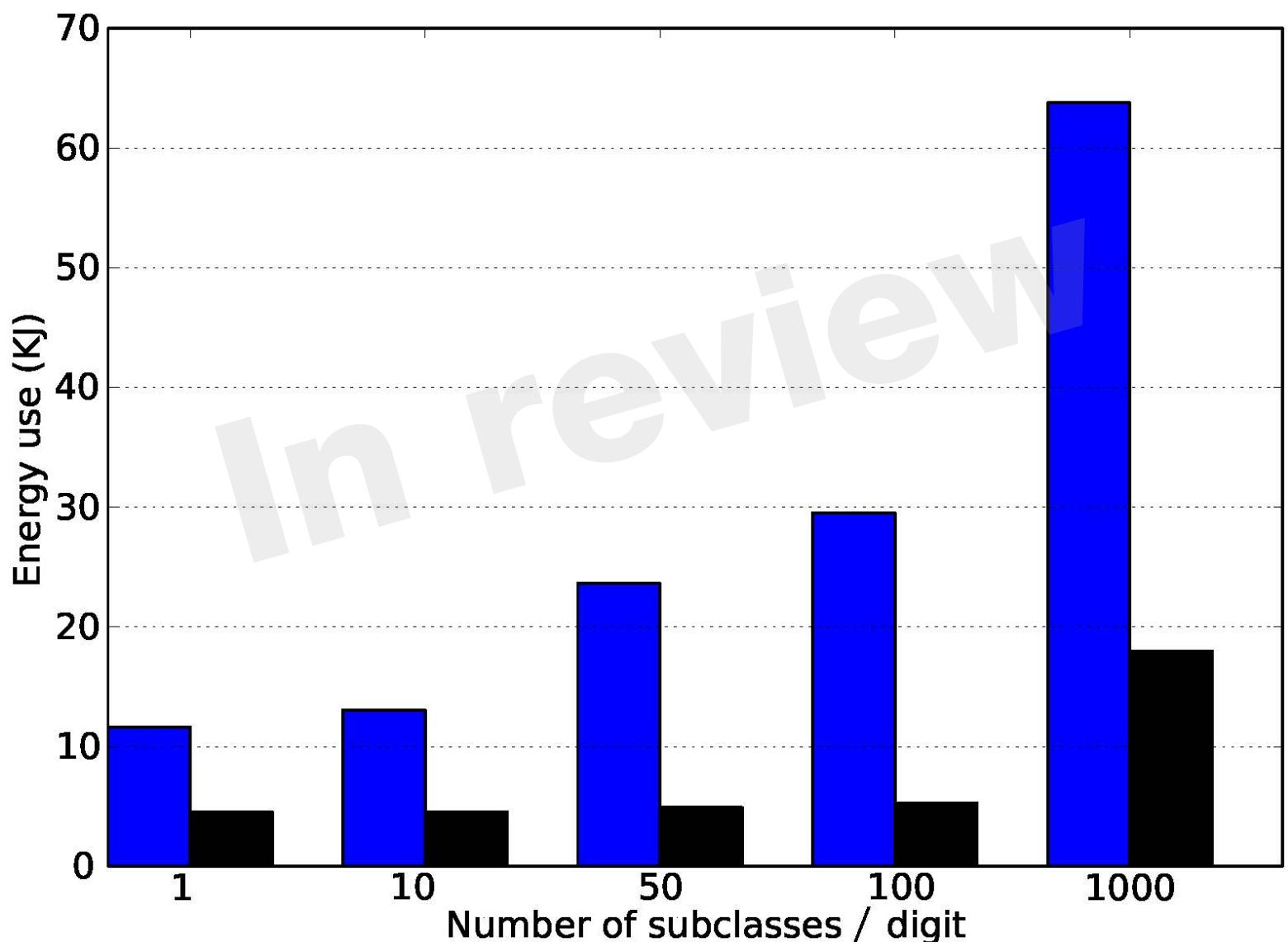


Figure 6.JPEG



Figure 7.JPEG



Figure 8.JPEG



Figure 9.JPG

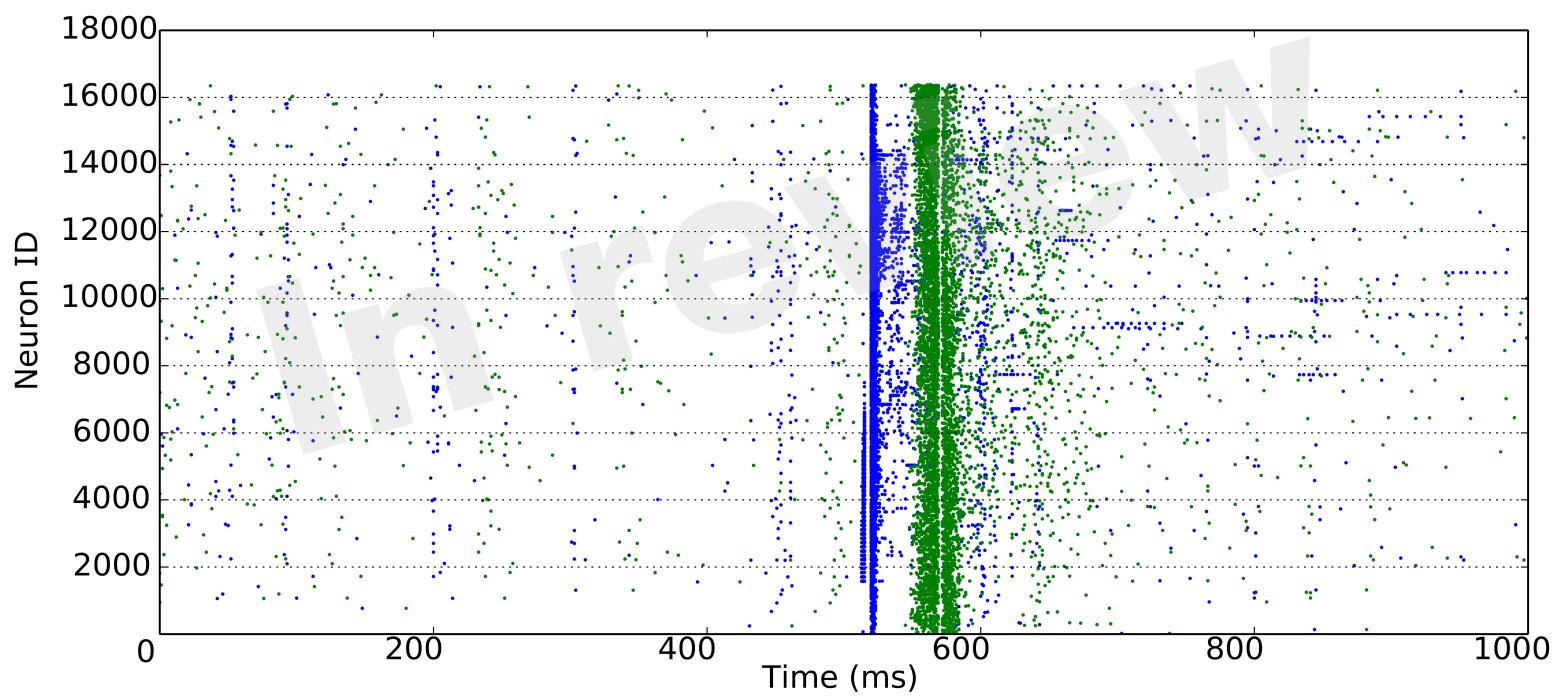


Figure 10.JPEG

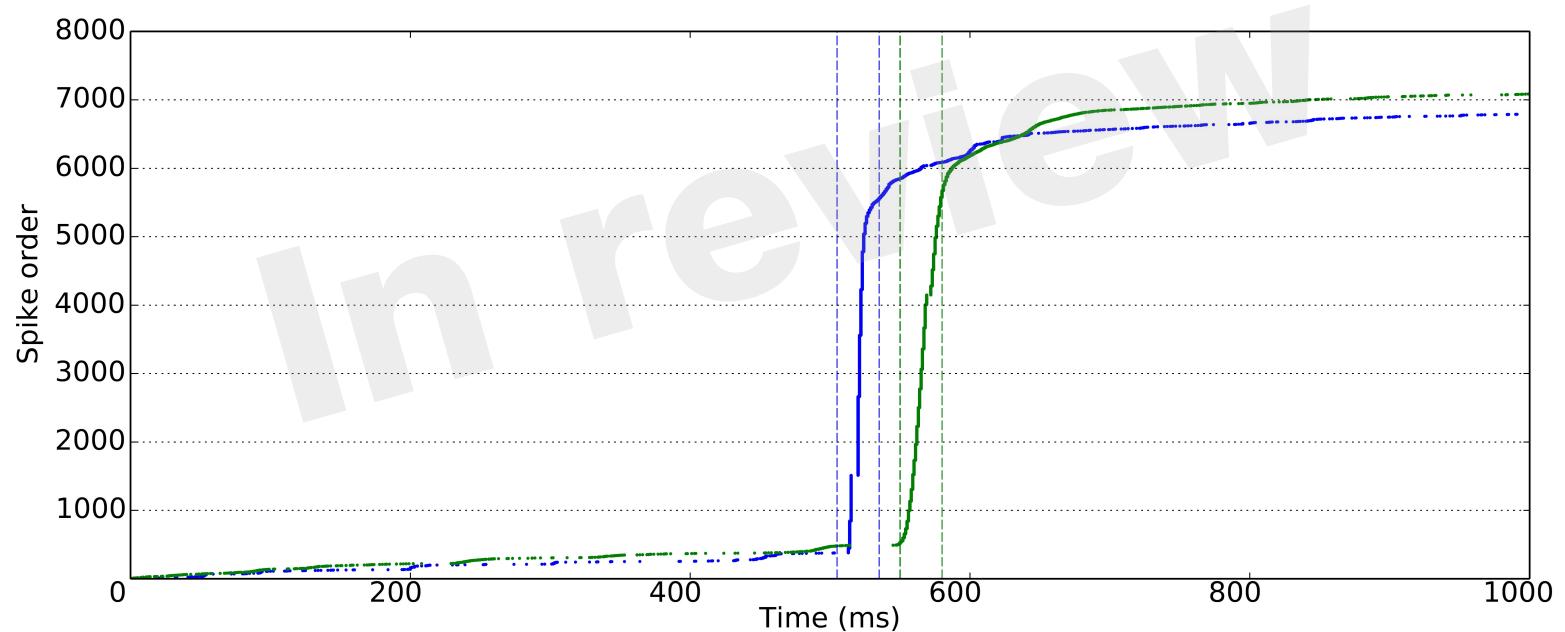


Figure 11.JPG

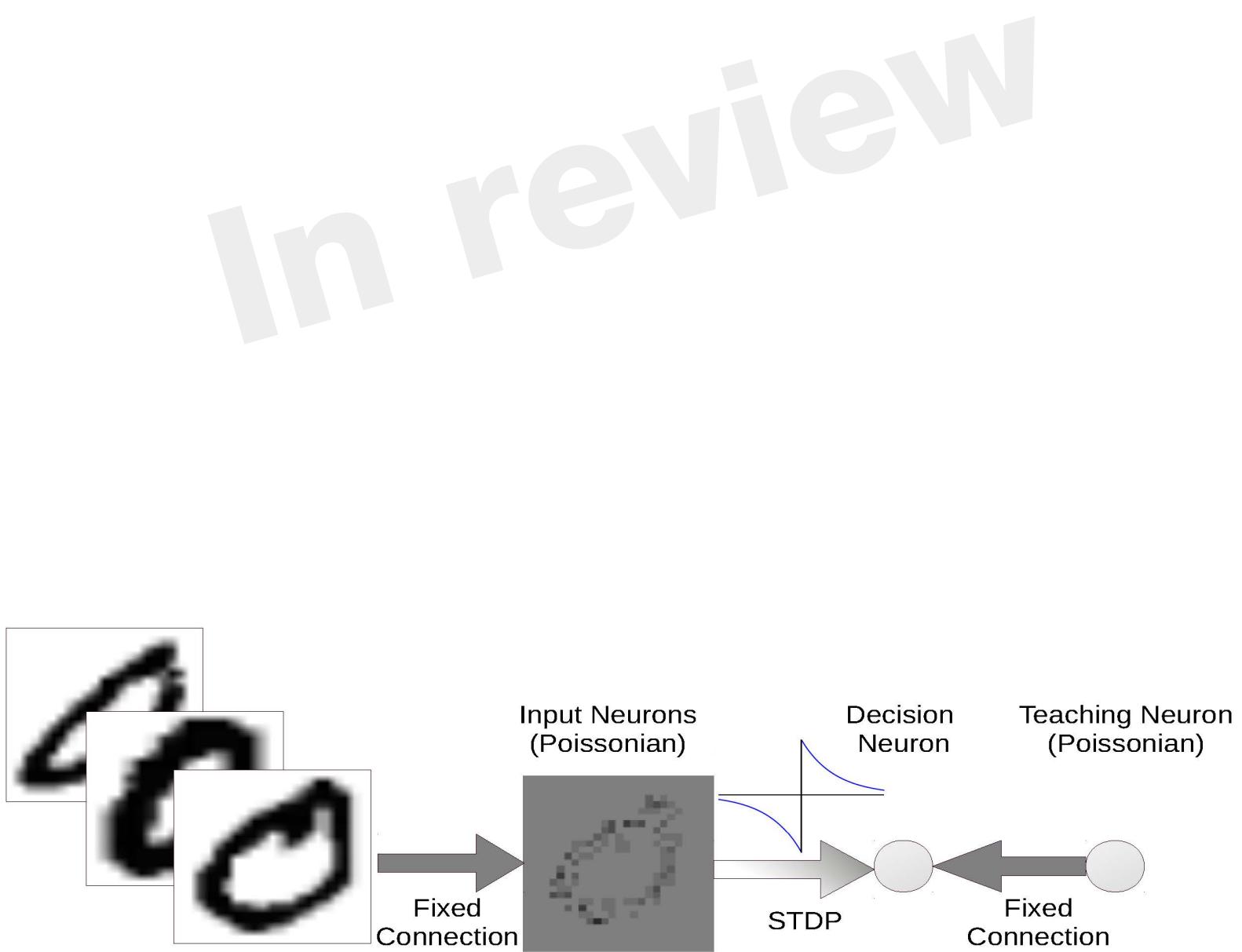


Figure 12.JPG

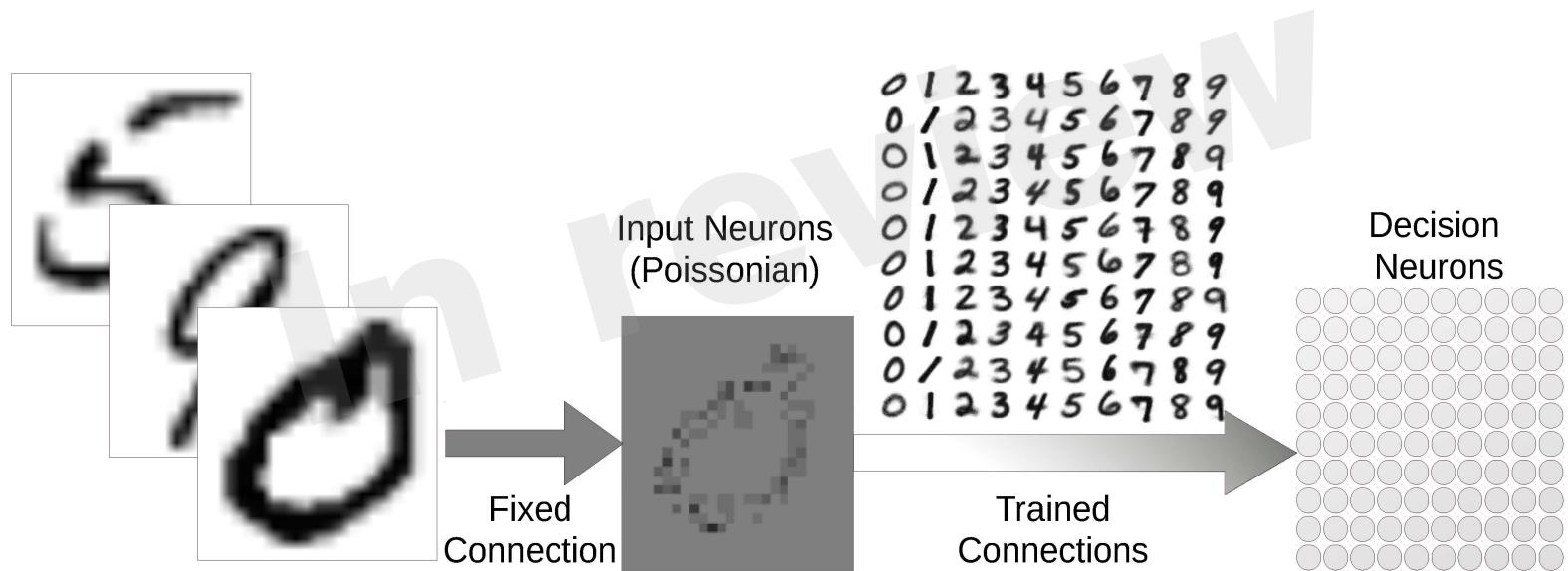


Figure 13.JPG

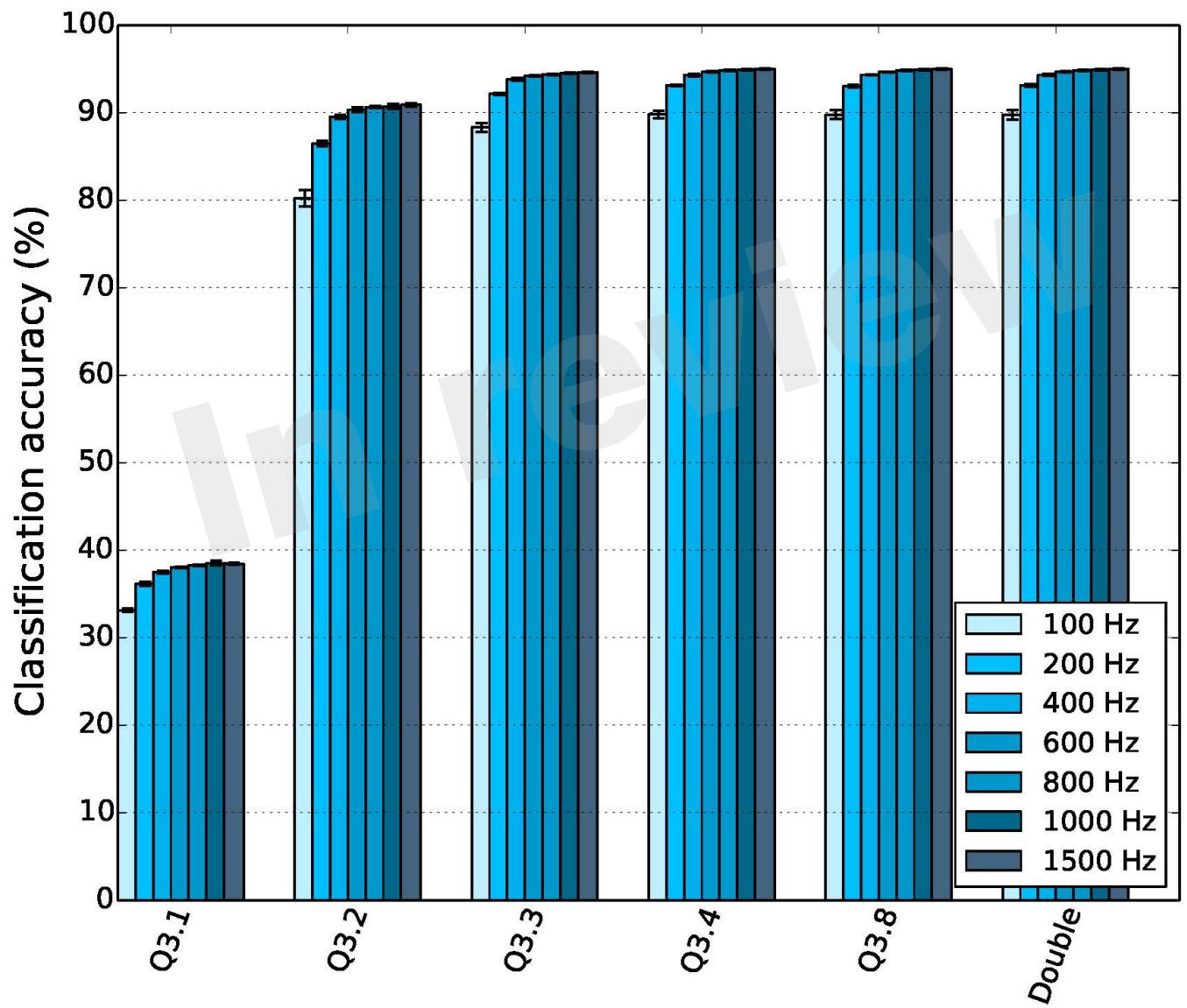


Figure 14.JPG

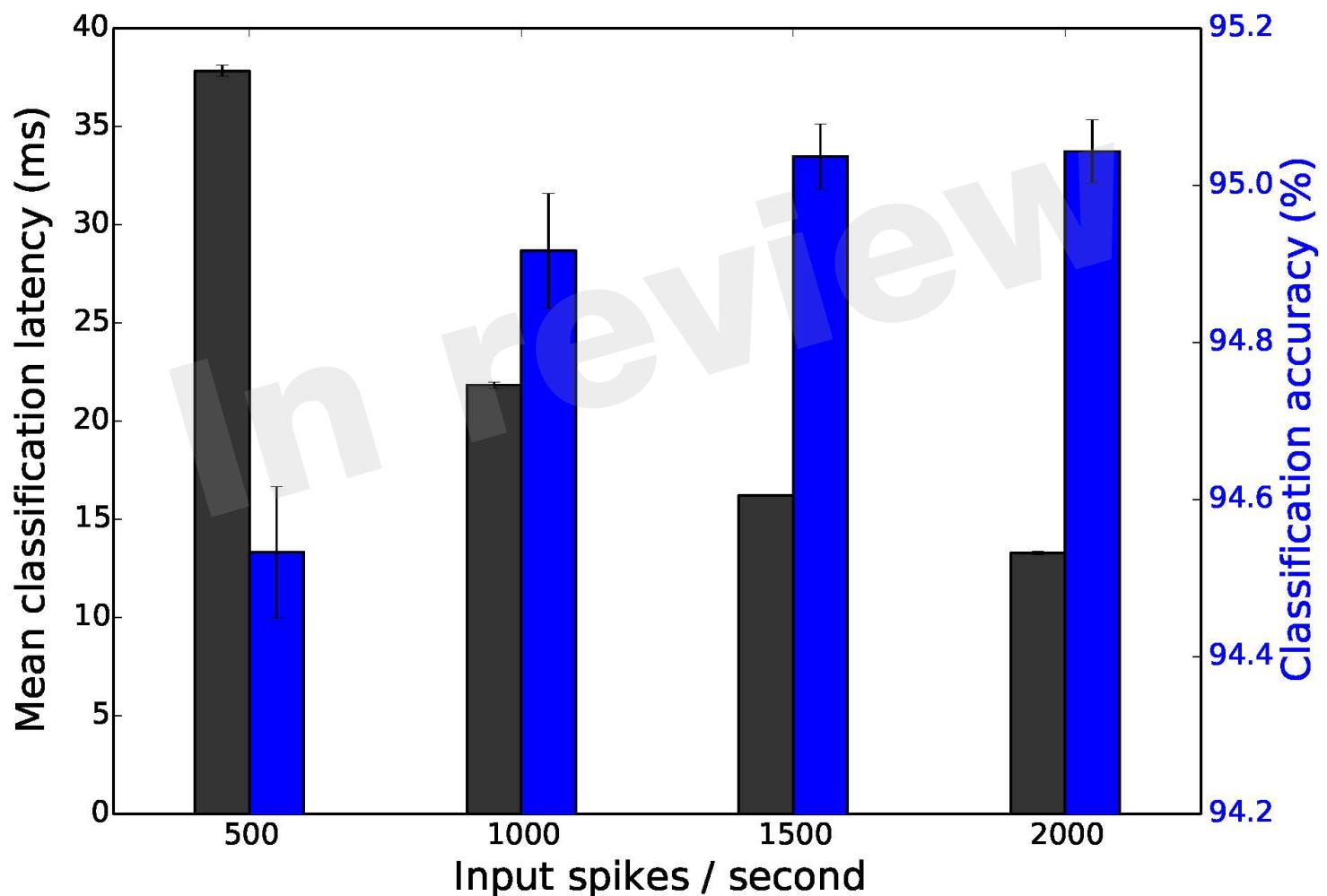


Figure 15.JPG

