VISION/GESTURE RECOGNITION USING SPIKING NEURONS

November 16, 2014

By
Qian Liu
School of Computer Science

Contents

List of Tables

List of Figures

Abstract

To explore how the brain may recognise objects in its general,accurate and energy-efficient manner, this paper proposes the use of a neuromorphic hardware system formed from a Dynamic Video Sensor (DVS) silicon retina in concert with the SpiNNaker real-time Spiking Neural Network (SNN) simulator. As a first step in the exploration on this platform a recognition system for dynamic hand postures is developed, enabling the study of the methods used in the visual pathways of the brain. Inspired by the behaviours of the primary visual cortex, Convolutional Neural Networks (CNNs) are modelled using both linear perceptrons and spiking Leaky Integrate-and-Fire (LIF) neurons.

In this study's largest configuration using these approaches, a network of 74,210 neurons and 15,216,512 synapses is created and operated in real-time using 290 SpiNNaker processor cores in parallel and with 93.0% accuracy. A smaller network using only 1/10th of the resources is also created, again operating in real-time, and it is able to recognise the postures with an accuracy of around 86.4% - only 6.6% lower than the much larger system. The recognition rate of the smaller network developed on this neuromorphic system is sufficient for a successful hand posture recognition system, and demonstrates a much improved cost to performance trade-off in its approach.

Chapter 1

Introduction

Patterns or objects in two-dimensional images can be described with four properties [1]: position, geometry (i.e. size, area and shape), colour/texture, and trajectory. Appearance-based methods are the most direct approach to performing pattern recognition where the test image is compared with a set of templates to find the best match for an individual or combination of properties. However, the 2D projection of an object changes under different conditions including illumination, viewing angles, relative positions and distance, making it virtually impossible to represent all appearances of an object. To improve reliability, robustness and classification efficiency, approaches such as edge matching [2], divide-and-conquer [3], gradient matching [4] and feature based methods [5, 6] are used. Finding a proper feature for a specific object still remains an open question and there is no process as general, accurate, or energy-efficient as that provided by the brain. It is not a new idea to turn to nature for inspiration. Riesenhuber et al. [7], for instance, presented a biologically-inspired model based on the organisation of the visual cortex which has the ability to represent relative position- and scale-invariant features. Integrating a rich set of visual features became possible using a feed-forward hierarchical pathway.

1.1 Aim

To explore how brain may recognise objects, we have employed a biologically-inspired DVS silicon retina [8], a good example of low-cost visual processing due to its event-driven and redundancy-reducing style of computation; and a SpiNNaker system [9], which is a massive parallel computing platform aimed at real-time simulation of SNNs.

With this neuromorphic hardware system we have the ability to explore visual processing by mimicking the functions of various regions along the visual pathway. Building a real-time recognition system for dynamic hand postures is a first step of exploring visual processing in a biological fashion and is also a validation of the neuromorphic platform. To match the image properties detailed earlier, the position, shape, size and trajectory of the hand postures can be detected from the retina output. To keep the task simple at first, the postures are of similar size and the goal is to recognise the shape of a hand with moving positions. Tracking the postures with a short memory will form part of the future work.

1.2 Why is it important

Why is it important to research on vision process in the brain.

Chapter 2

Background

2.1 Posture/Gesture Recognition

Dynamic recognition takes advantage of the intrinsic temporal processing of SNNs which are receiving considerable attention for undertaking vision processing. Pattern information can be encoded in the delays between the pre- and post-synaptic spikes since the spiking neurons are capable of computing radial basis functions (RBFs) [10]. Spatio-temporal information can also be stored in the exact firing time rather than relative delays [11]. Maass [12] has proved mathematically that: 1) networks of spiking neurons are computationally more powerful than the first and second generation of neural network models; 2) a concrete biologically relevant function can be computed by a single spiking neuron, replacing hundreds of hidden units in a sigmoidal neural net; 3) any function that can be computed by a small sigmoidal neural net can also be computed by a small network of spiking neurons. Numerous applications using SNN-based vision processing have been successfully carried out in the past. A dual-layer SNN has been trained using Spike Time Dependent Plasticity (STDP) and employed for character recognition [13]. Lee et al. [14] have implemented direction selective filters in real time using spiking neurons, considered as a convolution layer in the model of a so called CNN [15]. Different features, such as Gabor filter features (scale, orientation and frequency) and shape can be modelled as layers of feature maps. The similar behaviours have been found in the primary visual cortex (V1) in the visual pathway [16] as the foundation for higher level visual process e.g. object recognition. Rank order coding, as an alternative to conventional rate-based coding, treats the first spike as the most important and has been successfully applied to an orientation detection training process [17]. Nengo [18] is a graphical and scripting based

software package for simulating large-scale neural systems and has been used to build the world's largest functional brain model, Spaun [19]. An FPGA implementation of a Nengo model for digit recognition has been reported [20]. Deep Belief Networks (DBNs), the 4th generation of artificial neural network, have shown great success in solving classification problems. Recent study [21] in this area has mapped an offline-trained DBN onto an efficient event-driven spiking neural network for digit recognition tasks with resounding success.

2.2 Biology Aspect

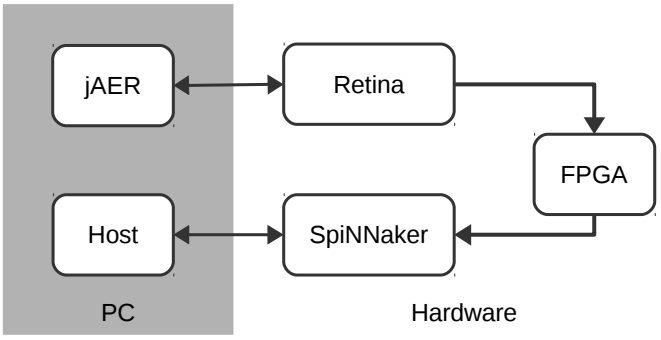
A lot to work on this part. May or may not include neuron models.

2.3 Platforms

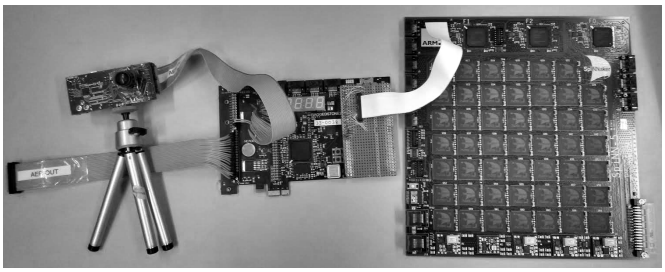
The outline of the platform is illustrated in Figure 2.1a, where the hardware system is configured, controlled and monitored by the PC. The jAER [22] event-based processing software on the PC configures the retina and displays the output spikes through a USB link. The host communicates to the SpiNNaker board via Ethernet to set up its runtime parameters and to download the neural network model off-line. It visualises [23] the spiking activity of the network in real-time. The photograph of the hardware platform, Figure 2.1b, shows that the silicon retina connects to the SpiNNaker 48-node system via a Spartan-6 FPGA board [24].

2.3.1 Vision Processing Front-ends

The visual input is captured by a DVS silicon retina, which is quite different from conventional video cameras. Each pixel generates spikes when its change in brightness reaches a defined threshold. Thus, instead of buffering video into frames, the activity of pixels is sent out and processed continuously with time. The communication bandwidth is therefore optimised by sending activity only, which is encoded as pixel events using Address-Event Representation (AER [25]) protocol. The level of activity depends on the contrast change; pixels generate spikes faster and more frequently when they are subject to more active change. The sensor is capable of capturing very fast moving objects (e.g., up to 10 K rotations per second), which is equivalent to 100 K conventional frames per second [8].



(a) Outline of the platform.



(b) Picture of the hardware platform. From left to right: a silicon retina, a FPGA board, and a 48-node SpiNNaker system.

Figure 2.1: System overview of the dynamic hand posture recognition platform.

2.3.2 SNNs Back-ends

The SpiNNaker project's architecture mimics the human brain's biological structure and functionality. This offers the possibility of utilizing massive parallelism and redundancy, as the brain, to provide resilience in an environment of unreliability and failure of individual components.

In the human brain, communication between its computing elements, or neurons, is achieved by the transmission of electrical 'spikes' along connecting axons. The biological processing of the neuron can be modelled by a digital processor and the axon connectivity can be represented by messages, or information packets, transmitted between a large number of processors which emulate the parallel operation of the billions of neurons comprising the brain.

The engineering of the SpiNNaker concept is illustrated in Figure 2.2 where the hierarchy of components can be identified. Each element of the toroidal interconnection mesh is a multi-core processor known as the 'SpiNNaker Chip' comprising 18 processing cores. Each core is a complete processing sub-system with local memory. It is

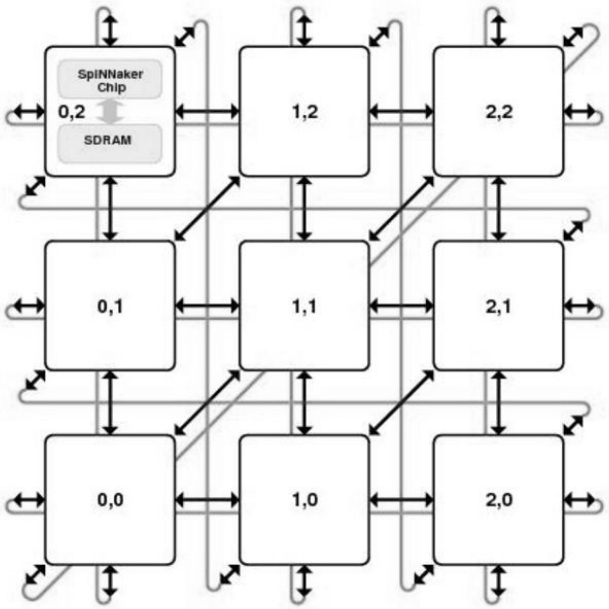


Figure 2.2: SpiNNaker system diagram. Each element represents one chip with local memory. Every chip connects to its neighbours through the six bi-directional on-board links.

connected to its local peers via a Network-on-Chip (NoC) which provides high bandwidth on-chip communication and to other SpiNNaker chips via links between them. In this way massive parallelism extending to thousands or millions of processors is possible.

The '103 machine' is the name given to the 48-node board which we use for the hand posture recognition system, see Figure 2.3. It has 864 ARM processor cores, typically deployed as 768 application, 48 monitor and 48 spare cores. The boards can be connected together to form larger systems using high-speed serial interfaces.

2.3.3 SpiNNaker distinguishing features

Spikes from the silicon retina are injected directly into SpiNNaker via a SPARTAN-6 FPGA board that translates them into a SpiNNaker compatible AER format [26].

From a neural modelling point of view, interfacing the silicon retina is performed using pyNN [27]. The retina is configured as a spike source population that resides on a virtual SpiNNaker chip, to which an AER sensor's spikes are directed, thus abstracting away the hardware details from the user[24]. Besides the retina, we have successfully connected an AER based silicon cochlea [28] to SpiNNaker for a sound localisation

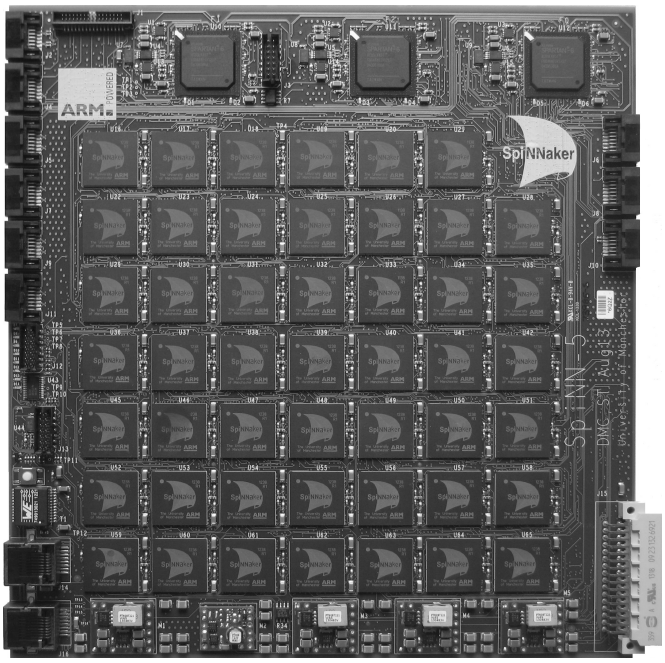


Figure 2.3: ‘103 Machine’ PCB

task [29].

2.4 Convolutional Neural Networks

The convolutional network is well-known as an example of a biologically-inspired model. Figure 2.4 shows a typical convolutional connection between two layers of neurons. The repeated convolutional kernels are overlapped in the receptive fields of the input neurons.

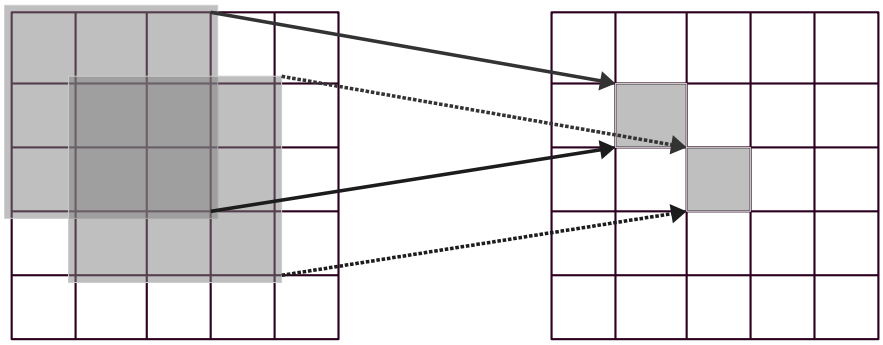


Figure 2.4: Each individual neuron in the convolution layer (right matrix) connects to its receptive field using the same kernel. The value of the kernel is represented by the synaptic weights between the connected neurons.

Chapter 3

CNNs Models

There are two CNNs proposed to accomplish the dynamic hand posture recognition task. A straight forward method of template matching is employed at first, followed by a network of multi-layer perceptrons (MLP) trained to improve the recognition performance.

Model 1: Template Matching. Shown in Figure 3.1 the first layer is the retina input, followed by the convolutional layer, where the kernels are Gabor filters responding to edges of four orientations. The third layer is the pooling layer where the size of the populations shrinks. This down-sampling enables robust classification due to its tolerance to variations in the precise shape of the input. The fourth layer is another convolution layer where the output from the pooling layer is convolved with the templates. The optional layer of Winner-Take-All (WTA) neurons enables a clearer classification result due to the inhibition between the neurons. In the Matlab simulation, the retina input spikes are buffered into 30 ms frames, and the neurons are simple linear perceptrons. The templates are chosen by sampling the output of the pooling layer when given some reference stimulus, see Figure 3.2.

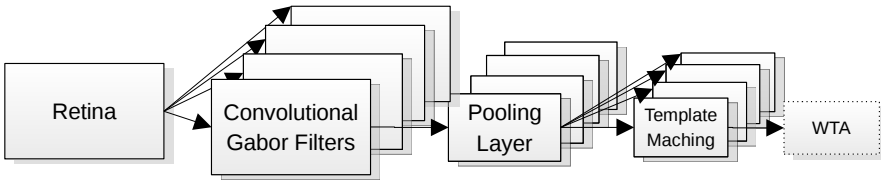


Figure 3.1: Model 1. The retina input is convolved with Gabor filters in the second layer, and then shrinks the sizes in the pooling layer. The templates are considered as convolution kernels in the last layer. The WTA circuit can be used as an option to show the template matching result more clearly.

Figure 3.2: Templates of the five postures: 'Fist','Index Finger', 'Victory Sign', 'Full Hand' and 'Thumb up'.

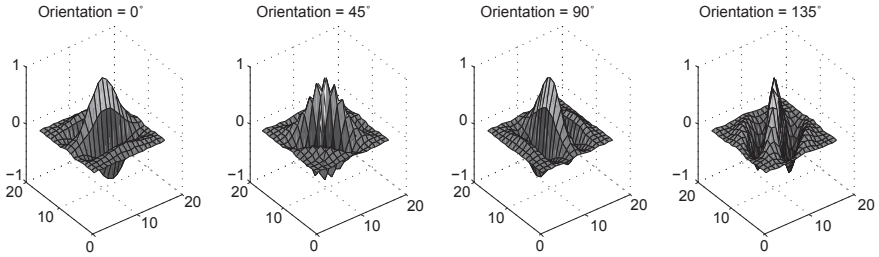


Figure 3.3: Real parts of the Gabor filters orienting four directions.

The Gabor filter is well-known as a linear filter for edge detection in image processing. A Gabor filter is a 2D convolution of a Gaussian kernel function and a sinusoidal plane wave; see Equation 3.1.

$$\begin{aligned}
&\text{RealParts} = \exp\left(\frac{-x'^2+y'^2}{2\sigma^2}\right)\cos\left(2\pi\frac{x'}{\lambda}\right)\\
&\text{ImaginaryParts} = \exp\left(\frac{-x'^2+y'^2}{2\sigma^2}\right)\sin\left(2\pi\frac{x'}{\lambda}\right)\\
&\text{where :}\\
&x' = x\cos(\theta)+y\sin(\theta)\\
&y' = -x\sin(\theta)+y\cos(\theta)
\end{aligned} \tag{3.1}$$

θ represents the orientation of the filter, λ is the wavelength of the sine wave, and σ is the standard deviation of the Gaussian envelope. The frequency and orientation features are similar to the responses of V1 neurons in the human visual system. Only the real parts of the Gabor filters (see Figure 3.3) are used as the convolutional kernels to configure the weights between the input layer and the Gabor filter layer.

The output score of a convolution is determined by the matching degree between

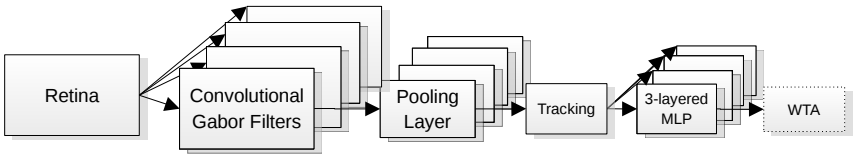


Figure 3.4: Model 2. The retina input convolves with Gabor filters in the second layer, and then shrinks the sizes in the pooling layer. The following tracking layer finds the most active area of some fixed size, moves the posture to the centre and pushes the image to the trained MLP. The winner-take-all (WTA) layer can be used as an option to show the template matching result more clearly.

the input and the kernel. Regarding the template matching layer, each neuron in a population responds to how closely its receptive field matches the specific template. The position of moving gesture is also naturally encoded in the address of template matching neuron. Thus, there are five populations of template matching neurons, one for each hand posture listed.

Model 2: Trained MLP. Inspired by the research of Lecun [30], we designed a combined network model with MLP and CNN (Figure 3.4). The first three layers are exactly the same as the previous model. The training images for the 3-layered MLP are of same size and the posture is centred in the images. Therefore, a tracking layer plays an important role to find the most active region and forward the centred image to the next layer.

3.0.1 Experimental Set-up

In order to evaluate the cost and performance trade-offs in optimizing the number of neural components, both the convolutional models described above are tested at different scales. Five videos of every posture are captured from the silicon retina in AER format, all of similar size and moving clock-wise in front of the retina. The videos are cut into frames (30 ms per frame) and presented to the convolutional networks. The configurations of the networks are listed in Table 3.1. The integration layer is not necessary in a convolutional network, but is used here to decrease the number of synaptic connections.

3.0.2 Experimental Results

In Figure 3.5 the first two plots refer to Model 1, using template matching. Each colour represents one of the recognition populations. Each point in the plot is the

Table 3.1: Sizes of the convolutional neural networks.

(a) Model 1: Template matching

	Full Resolution 128 × 128		**Sub-sampled Resolution 32 × 32**	
	Population Size	Connections per Neuron	Population Size	Connections per Neuron
Retinal Input	128 × 128	1	32 × 32	4 × 4
Gabor Filter	112×112×4	17 × 17	28×28×4	5 × 5
Pooling Layer	36×36×4	5 × 5	null	null
Integration Layer	36 × 36	4	28 × 28	4
Template Matching	16×16×5	21 × 21	14×14×5	15 × 15
Total	74,320	15,216,512	5,925	318,420

(b) Model 2: Trained MLP

	Full Resolution 128 × 128		**Sub-sampled Resolution 32 × 32**	
	Population Size	Connections per Neuron	Population Size	Connections per Neuron
Tracked Input	21 × 21	null	15 × 15	null
Hidden Layer	10	21×21×10	10	15×15×10
Recognition Layer	5	5×10	5	5×10
Total	456	4,460	240	2,300

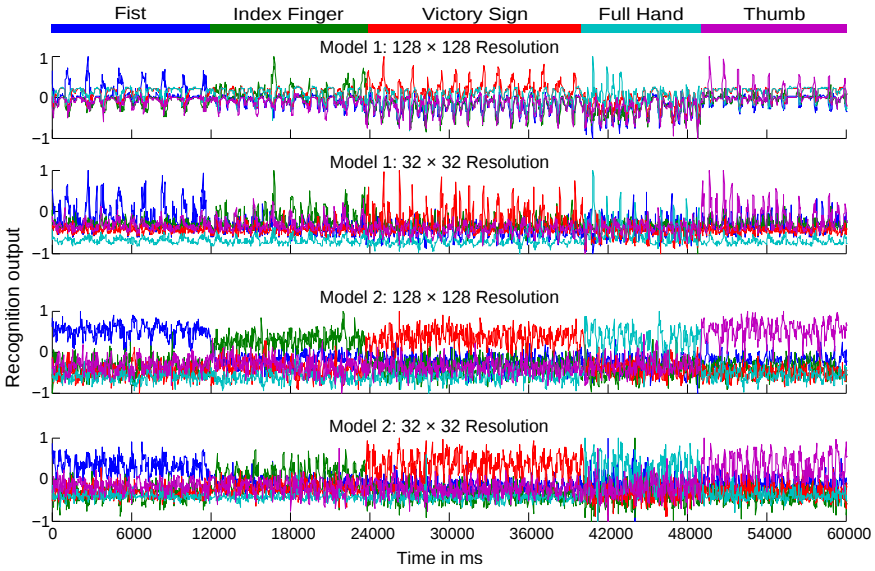


Figure 3.5: Neural responses with time of four experiments to the same recorded moving postures. The recognition output is normalised to [-1, 1]. Every point represents the highest response in a specific population (different colour) for a 30 ms frame. The 1st plot refers to Model 1 with the full input resolution, and the 2nd plot Model 1 with the sub-sampled input resolution; and the 3rd and fourth plots both refer to Model 2, and with high and low input resolution respectively.

highest neuronal response in the recognition population during the time of one frame (30 ms). The neuronal response, 'the spiking rate', is normalised to [-1, 1]. It can be seen that the higher resolution input makes the boundaries between the classes clearer. On the other hand, recognition only happens when the test image and template are similar enough. The templates are only selected from the frames where the gestures are moving towards the right, and the gestures are moving clockwise in the videos, thus, all the peaks in plot 1 correspond with moments when the gesture moves towards right. It is notable that the higher resolution causes the recogniser to be more sensitive to the differences between the test data and the template, while the smaller neural network can recognize more generalized patterns. Therefore, a threshold is required to differentiate between data that is close enough and that which is not. Since the gestures are moving in four different directions during the clockwise movement, a rejection rate (i.e. none of the template is matched) of 75% is to be expected.

The latter two plots of Figure 3.5 refer to Model 2. The three-layer MLP network significantly improves the recognition rate and can generalise the pattern. There is no rejection rate for Model 2, since the MLP is trained with all the moving directions of

Table 3.2: Recognition results using linear perceptrons in %

		Model 1		Model 2	
		High Resolution	Low Resolution	High Resolution	Low Resolution
Fist	Correct	99.11	99.23	96.24	84.21
(399 Frames)	Reject	71.93	67.42	Null	Null
Index Finger	Correct	92.98	80.00	94.39	71.69
(392 Frames)	Reject	70.92	75.77	Null	Null
Victory Sign	Correct	96.56	93.07	95.64	87.66
(551 Frames)	Reject	73.68	81.67	Null	Null
Full Hand	Correct	95.65	72.41	93.52	72.01
(293 Frames)	Reject	92.15	90.10	Null	Null
Thumb up	Correct	89.61	84.44	96.68	74.68
(391 Frames)	Reject	80.31	76.98	Null	Null
Average	Correct	94.78	85.83	95.29	78.05
	Reject	77.80	78.39	Null	Null

the postures.

Detailed results are listed in Table 3.2. The correct recognition rate is calculated from the non-rejected frames. The lower resolution of the 32×32 retina input is adequate (85.83%) for this gesture recognition task. The smaller network uses only 1/10th the number of neurons and 1/50th the number of synaptic connections compared with the full resolution network, while the recognition rate drops only around by 9.0% with Model 1 and 17.2% with Model 2.

Chapter 4

Recognition on SpiNNaker

4.0.3 Moving from Rate-based Perceptrons to Spiking Neurons

It remains a challenge to transform traditional artificial neural networks into spiking ones. There are attempts [31] [32] to estimate the output firing rate of the LIF neurons (Equation 4.1) under certain conditions.

$$\frac{\mathrm{d}V(t)}{\mathrm{d}t} = -\frac{V(t) - V_{rest}}{\tau_m} + \frac{I(t)}{C_m} \tag{4.1}$$

The membrane potential V changes in response to input current I, starting at the resting membrane potential V_{rest}, where the membrane time constant is $\tau_m = R_m C_m$, R_m is the membrane resistance and C_m is the membrane capacitance.

Given a constant current injection I, the response function, i.e. firing rate, of the LIF neuron is

$$\lambda_{out} = \left[t_{ref} - \tau_m \ln\left(1 - \frac{V_{th} - V_{rest}}{IR_m}\right)\right]^{-1} \tag{4.2}$$

when $IR_m > V_{th} - V_{rest}$, otherwise the membrane potential cannot reach the threshold V_{th} and the output firing rate is zero. The absolute refractory period t_{ref} is included, where all input during this period is invalid. In a more realistic scenario, the post-synaptic potentials (PSPs) are triggered by the spikes generated from the neuron's pre-synaptic neurons other than a constant current. Assume that the synaptic inputs are Poisson spike trains, the membrane potential of the LIF neuron is considered as a diffusion process. Equation 4.1 can be modelled as a stochastic differential equation referring to Ornstein-Uhlenbeck process,

$$\tau_m \frac{\mathrm{d}V(t)}{\mathrm{d}t} = -\left[V(t) - V_{rest}\right] + \mu + \sigma\sqrt{2\tau_m}\xi(t) \tag{4.3}$$

where

$$\mu = \tau_m(\mathbf{w_E} \cdot \lambda_\mathbf{E} - \mathbf{w_I} \cdot \lambda_\mathbf{I})$$
$$\sigma^2 = \frac{\tau_m}{2}\left(\mathbf{w_E^2} \cdot \lambda_\mathbf{E} + \mathbf{w_I^2} \cdot \lambda_\mathbf{I}\right) \tag{4.4}$$

are the conditional mean and variance of the membrane potential. The delta-correlated process $\xi(t)$ is Gaussian white noise with zero mean, $\mathbf{w_E}$ and $\mathbf{w_I}$ stand for the weight vectors of the excitatory and the inhibitory synapses, and λ represents the vector of the input firing rate. The response function of the LIF neuron with Poisson input spike trains is given by the Siegert function [33],

$$\lambda_{out} = \left(\tau_{ref} + \frac{\tau_Q}{\sigma_Q}\sqrt{\frac{\pi}{2}}\int_{V_{rest}}^{V_{th}} du \exp\left(\frac{u-\mu_Q}{\sqrt{2}\sigma_Q}\right)^2 \left[1 + erf\left(\frac{u-\mu_Q}{\sqrt{2}\sigma_Q}\right)\right]\right)^{-1} \tag{4.5}$$

where τ_Q, μ_Q, σ_Q are identical to τ_m, μ, σ in Equation 4.4, and erf is the error function.

Still there are some limitations on the response function. For the diffusion process, only small amplitude (weight) of the PostSynaptic Potentials (PSPs) generated by a large amount of input spikes (high spiking rate) work under this circumstance; plus, the delta function is required, i.e. the synaptic time constant is considered to be zero. Thus only a rough approximation of the output spike rate has been determined. Secondly, given different input spike rate to each pre-synaptic neurons, the parameters of the LIF neuron and the output spiking rate, how to tune every single corresponding synaptic weight remains a difficult task.

4.0.4 Live Recognition

We implemented the prototype of the dynamic posture recognition system on SpiNNaker using LIF neurons. The input retina layer consists of 128×128 neurons; each Gabor filter has 112×112 valid neurons, since the kernel size is 17×17; each pooling layer is as big as 36×36, convolving with five template kernels (21×21); thus, the recognition populations are 16×16 neurons each. Altogether 74,320 neurons and 15,216,512 synapses, use up to 19 chips (290 cores) on a 48-node board, see Table 3.1a. Regarding the lower resolution of 32×32 retinal input, the network (Table 3.1b) consists of 5,925 neurons and 318,420 synapses taking up only two chips (31 cores) of the board.

Figure 4.1 shows snapshots of neural responses of some populations during real-time recognition. Figure 4.1a is a snapshot of the Gabor population which prefers the

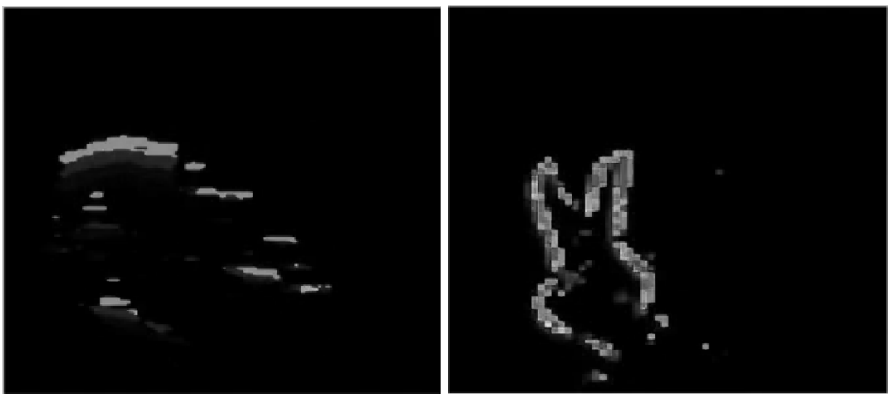
horizontal direction, given the input posture of a 'Fist'; and Figure 4.1b shows the activity of the neurons in the integration layer, given a 'Victory Sign'. And the active neurons in the visualiser in Figure 4.1c are pointing out the position of the recognised pattern the 'Index finger'. All the supporting demonstrative videos can be found on YouTube [34, 35, 36].

4.0.5 Recognition of Recorded Data

To compare with the results of the experiments carried out with Matlab (in Section 3.0.2), the same recorded retinal data is conducted into SpiNNaker. Only Model 1 is tested on the neuromorphic hardware platform, since tracking is still need to investigate using SNN (for Model 2) in the future. The recorded data is presented as spike source array in the system with 128×128 input (see Figure 4.3a) while the data is forwarded to a sub-sampling layer of 32×32 resolution in the system of the smaller network (see Figure 4.4a). The output spikes generated from the recognition populations with time are shown in Figures 4.3 and 4.4 for full resolution and lower systems respectively. More spikes are generated during the period when the preferred input posture is shown.

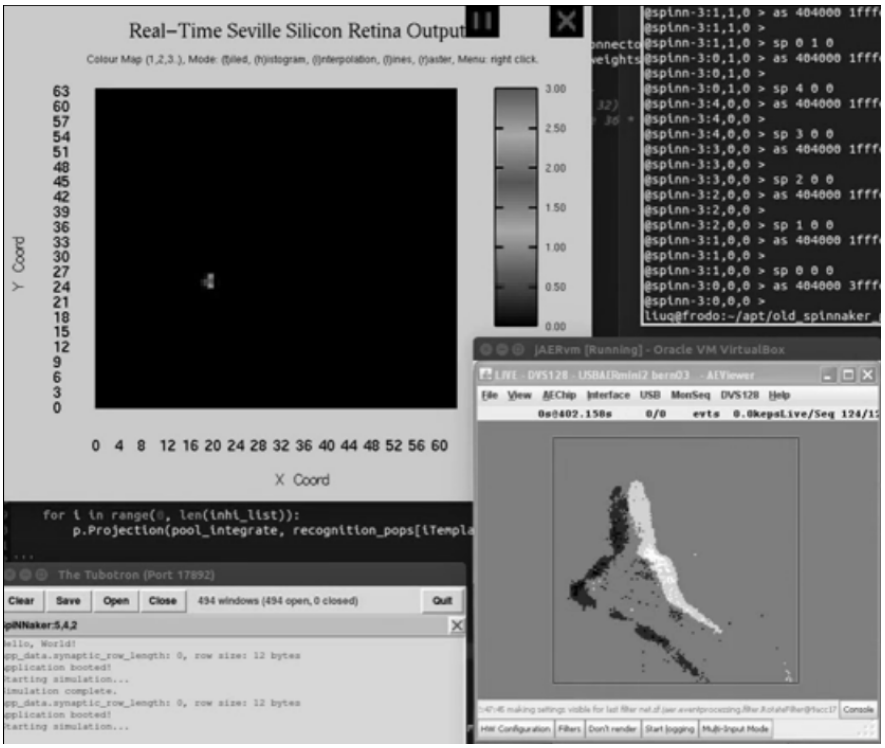
Correspondingly, the spiking rates of each recognition population is sampled into frames (Figure 4.2) to make a comparison with the Matlab simulation. Each colour represents one recognition population, and the spike activity goes higher when the input posture matches the template. Firstly, the spike rates are sampled into 30 ms frames which is in accordance with the Matlab experiments. In the Matlab simulation, the templates are trained with cut frames and so the test images are also fixed to the same length frames. Otherwise, the recogniser will not work properly because of the replications of the moving posture. Contrasting this, the spiking rates can be sampled to various frame lengths. Thus, the other two plots in the figure illustrate the classification in a wider window of 300 ms. From Table 4.1, the recognition and rejection rates are quantified as percentages.

Comparing with the results of Matlab simulation (Table 3.2), the recognition rate is about 7.6% lower at both high and low resolutions, and the rejection rate remains the same slightly above 75%. However, by changing the frame length to 300 ms recognition rates reach (93.0% for the larger network) or exceed (86.4% for smaller network) the Matlab simulation, meanwhile the rejection rates also drop dramatically by 26.0% and 22.4%. This is in accordance with natural visual responses, which means, the longer an object shows, the more accurate the recognition will be. Between the two network scales there is also a smaller gap in recognition rates as the window length



(a) Neural responses of the Gabor filter layer orienting to the horizontal direction [34]

(b) Neural responses of the integrate layer [35]



(c) Snapshot of the neuron responses of the template matching layer [36]

Figure 4.1: Snapshots of the real-time dynamic posture recognition system on SpiNNaker.

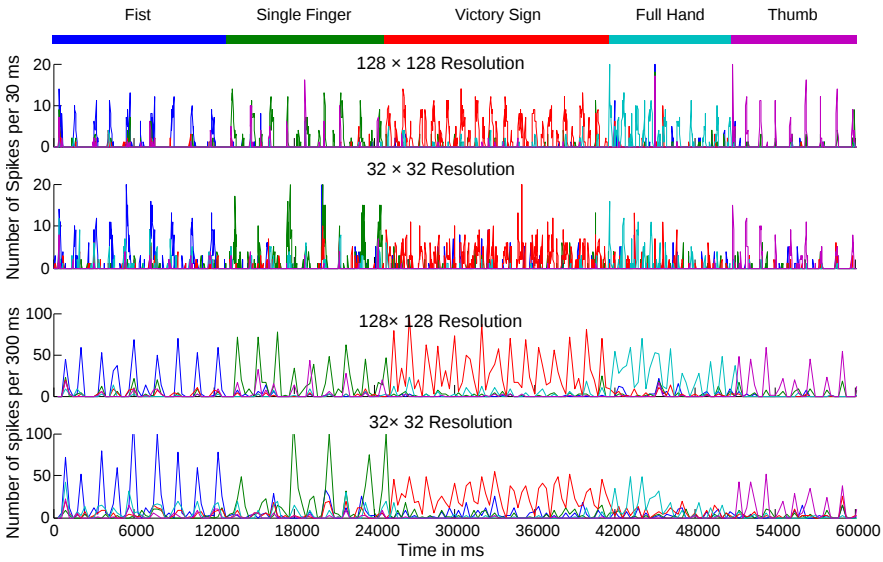
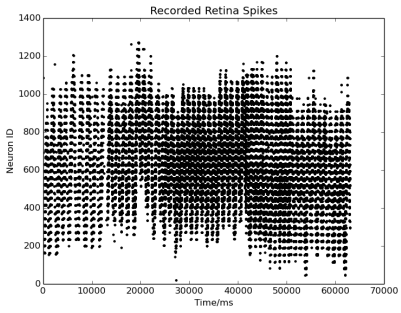


Figure 4.2: Real-time neural responses of two experiments on SpiNNaker with time to the same recorded postures. These two experiments only differ in input resolution. The result of the high input resolution test is plotted the first with a sample frame of 30 ms; while the 3rd plot shows the same result with a sample frame of 300 ms. The other two plots refer to the smaller input resolution. Every point represents the over all number of spikes of a specific population (different colour) in a 'frame'.

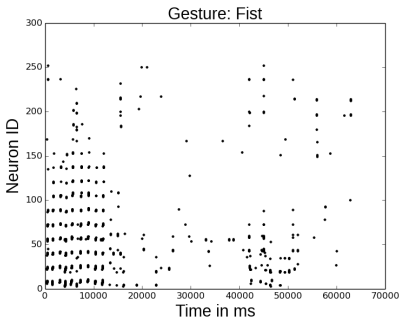
Table 4.1: Real-time recognition results on SpiNNaker in %

		30 ms per frame		**300 ms per frame**	
		High Resolution	Low Resolution	High Resolution	Low Resolution
Fist	Correct	91.78	78.02	100	92.31
	Reject	82.78	78.54	70.73	68.29
Index Finger	Correct	78.25	78.25	88.24	72.22
	Reject	80.46	73.56	57.50	55.00
Victory Sign	Correct	96.48	86.27	95.00	92.50
	Reject	64.46	72.68	28.57	28.57
Full Hand	Correct	85.29	60.78	90.00	75.00
	Reject	67.31	83.65	35.48	61.29
Thumb up	Correct	84.09	88.10	91.67	100
	Reject	87.54	73.81	66.67	66.67
Average	Correct	87.18	78.28	92.98	86.41
	Reject	76.51	76.45	51.79	55.96

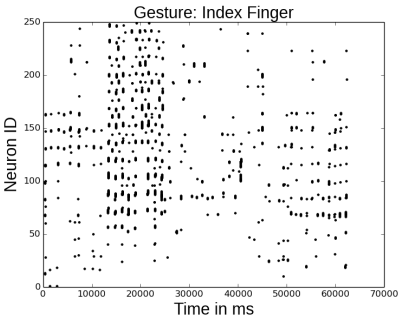
grows, i.e. 8.9% and 6.6% respectively.



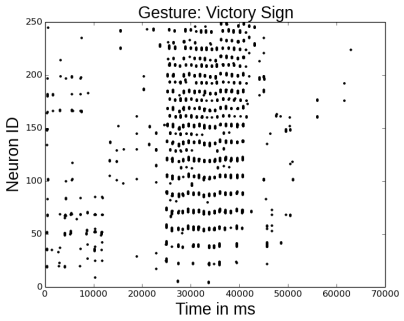
(a) Retinal input population



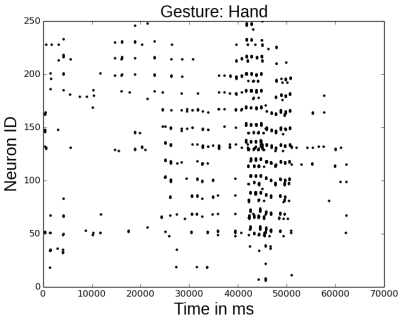
(b) Template matching population, 'Fist'



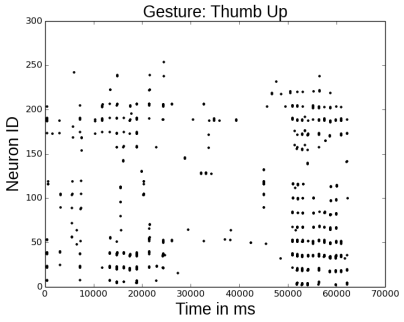
(c) Template matching population, 'Index Finger'



(d) Template matching population, 'Victory Sign'

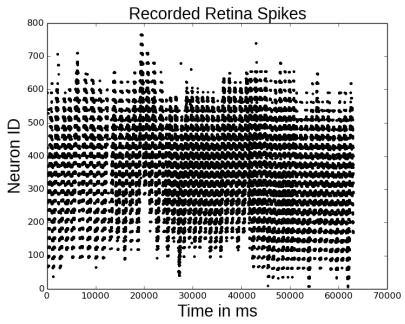


(e) Template matching population, 'Full Hand'

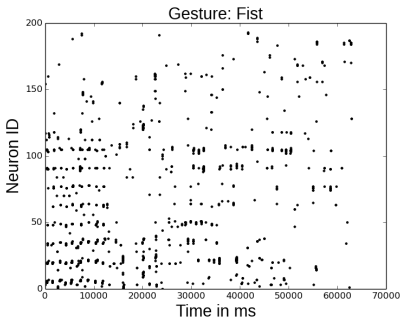


(f) Template matching population, 'Thumb Up'

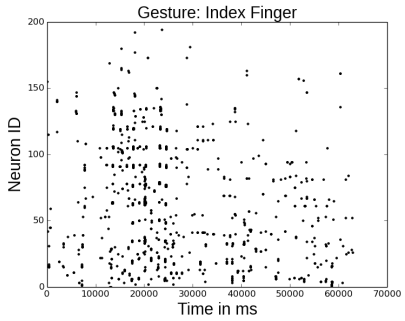
Figure 4.3: Spikes captured during the live recognition of the recorded retinal input with the resolution of 128×128.



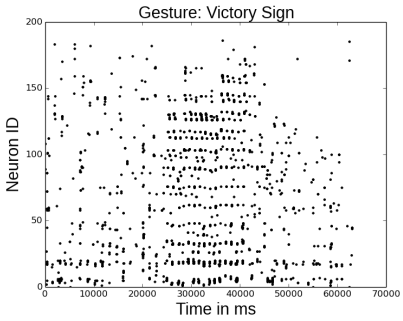
(a) Retinal input population



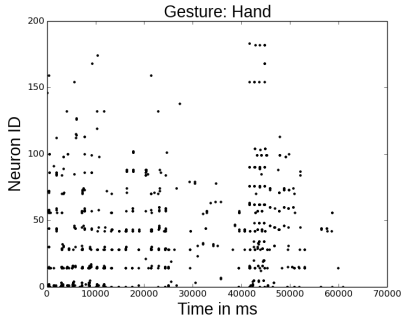
(b) Template matching population, 'Fist'



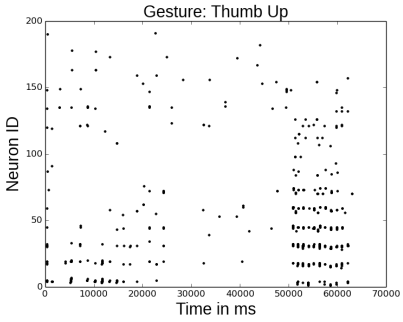
(c) Template matching population, 'Index Finger'



(d) Template matching population, 'Victory Sign'



(e) Template matching population, 'Full Hand'



(f) Template matching population, 'Thumb Up'

Figure 4.4: Spikes captured during the live recognition of the recorded retinal input with the resolution of 32×32.

Chapter 5

Research Plan

Chapter 6

Conclusion

To explore how brain may recognise objects in its general, accurate and energy-efficient manner, this paper proposes the use of a neuromorphic hardware system which includes a DVS retina connected to SpiNNaker, a real-time SNN simulator. Building a recognition system based on this bespoke hardware for dynamic hand postures is a first step in the study of visual pathway of the brain. Inspired by the structures of the primary visual cortex, convolutional neural networks are modelled using both linear perceptrons and LIF neurons. The larger network of 74,210 neurons and 15,216,512 synapses runs smoothly in real-time on SpiNNaker using 290 cores within a 48-node board. The smaller network using 1/10 of the resources is able to recognise the postures in real-time with an accuracy about 86.4% in average, which is only 6.6% lower than the former but with a better cost/performance ratio.

The future work on this topic will include further collaboration with biologists and neuroscientists working on vision systems, especially concentrating on the orientation detection region of the brain. To equip the system with tracking is another importance direction for future development where the recognition performance can be increased by exploiting short-term memory of a gesture's route. Using the approch of HMMs [37] and applying to spiking neural networks is an idea we wish to explore as part of this promising work.

Bibliography

[1] S. G. Wysoski, L. Benuskova, and N. Kasabov, "Fast and adaptive network of spiking neurons for multi-view visual pattern recognition," *Neurocomputing*, vol. 71, no. 13, pp. 2563–2575, 2008.

[2] J. Canny, "A computational approach to edge detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 6, pp. 679–698, 1986.

[3] Ö. Toygar and A. Acan, "Multiple classifier implementation of a divide-and-conquer approach using appearance-based statistical methods for face recognition," *Pattern Recognition Letters*, vol. 25, no. 12, pp. 1421–1430, 2004.

[4] S.-D. Wei and S.-H. Lai, "Robust and efficient image alignment based on relative gradient matching," *Image Processing, IEEE Transactions on*, vol. 15, no. 10, pp. 2936–2943, 2006.

[5] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[6] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[7] M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," *Nature neuroscience*, vol. 2, no. 11, pp. 1019–1025, 1999.

[8] J. A. Leñero-Bardallo, T. Serrano-Gotarredona, and B. Linares-Barranco, "A 3.6 s latency asynchronous frame-free event-driven dynamic-vision-sensor," *Solid-State Circuits, IEEE Journal of*, vol. 46, no. 6, pp. 1443–1455, 2011.

[9] S. B. Furber, F. Galluppi, S. Temple, and L. A. Plana, "The SpiNNaker Project," 2014.

[10] J. J. Hopfield, "Pattern recognition computation using action potential timing for stimulus representation," *Nature*, vol. 376, no. 6535, pp. 33–36, 1995.

[11] T. Natschläger and B. Ruf, "Spatial and temporal pattern analysis via spiking neurons," *Network: Computation in Neural Systems*, vol. 9, no. 3, pp. 319–332, 1998.

[12] W. Maass, "Networks of spiking neurons: the third generation of neural network models," *Neural networks*, vol. 10, no. 9, pp. 1659–1671, 1997.

[13] A. Gupta and L. N. Long, "Character recognition using spiking neural networks," in *Neural Networks, 2007. IJCNN 2007. International Joint Conference on*, pp. 53–58, IEEE, 2007.

[14] J. H. Lee, P. Park, C.-W. Shin, H. Ryu, B. C. Kang, and T. Delbruck, "Touchless hand gesture UI with instantaneous responses," in *Image Processing (ICIP), 2012 19th IEEE International Conference on*, pp. 1957–1960, Sept 2012.

[15] L. Camunas-Mesa, C. Zamarreno-Ramos, A. Linares-Barranco, A. J. Acosta-Jimenez, T. Serrano-Gotarredona, and B. Linares-Barranco, "An event-driven multi-kernel convolution processor module for event-driven vision sensors," *Solid-State Circuits, IEEE Journal of*, vol. 47, no. 2, pp. 504–517, 2012.

[16] M. Rehn and F. T. Sommer, "A network that uses few active neurones to code visual input predicts the diverse shapes of cortical receptive fields," *Journal of computational neuroscience*, vol. 22, no. 2, pp. 135–146, 2007.

[17] A. Delorme, L. Perrinet, and S. J. Thorpe, "Networks of integrate-and-fire neurons using rank order coding b: spike timing dependent plasticity and emergence of orientation selectivity," *Neurocomputing*, vol. 38, pp. 539–545, 2001.

[18] C. Eliasmith and T. C. Stewart, "Nengo and the neural engineering framework: connecting cognitive theory to neuroscience," in *Proceedings of the 33rd annual meeting of the cognitive science society*, pp. 1–2, 2011.

[19] C. Eliasmith, T. C. Stewart, X. Choo, T. Bekolay, T. DeWolf, Y. Tang, and D. Rasmussen, "A large-scale model of the functioning brain," *science*, vol. 338, no. 6111, pp. 1202–1205, 2012.

[20] M. Naylor, P. J. Fox, A. T. Markettos, and S. W. Moore, "Managing the fpga memory wall: Custom computing or vector processing?," in *Field Programmable Logic and Applications (FPL), 2013 23rd International Conference on*, pp. 1–6, IEEE, 2013.

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

[22] T. Delbruck, "Frame-free dynamic digital vision," in *Proceedings of Intl. Symp. on Secure-Life Electronics, Advanced Electronics for Quality Life and Society*, pp. 21–26, 2008.

[23] C. Patterson, F. Galluppi, A. Rast, and S. Furber, "Visualising large-scale neural network models in real-time," in *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pp. 1–8, 2012.

[24] F. Galluppi, K. Brohan, S. Davidson, T. Serrano-Gotarredona, J.-A. P. Carrasco, B. Linares-Barranco, and S. Furber, "A real-time, event-driven neuromorphic system for goal-directed attentional selection," in *Neural Information Processing*, pp. 226–233, Springer, 2012.

[25] J. Lazzaro and J. Wawrzynek, "A multi-sender asynchronous extension to the aer protocol," in *Advanced Research in VLSI, Conference on*, pp. 158–158, IEEE Computer Society, 1995.

[26] L. A. Plana, "AppNote 8 - Interfacing AER devices to SpiNNaker using an FPGA." `https://spinnaker.cs.man.ac.uk/tiki-download_wiki_attachment.php?attId=20`, 4 2013.

[27] A. P. Davison, D. Brüderle, J. Eppler, J. Kremkow, E. Muller, D. Pecevski, L. Perrinet, and P. Yger, "Pynn: a common interface for neuronal network simulators," *Frontiers in neuroinformatics*, vol. 2, 2008.

[28] S.-C. Liu, A. van Schaik, B. Minch, and T. Delbruck, "Event-based 64-channel binaural silicon cochlea with q enhancement mechanisms," in *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pp. 2027–2030, May 2010.

[29] Q. Liu, C. Patterson, S. Furber, Z. Huang, Y. Hou, and H. Zhang, "Modeling populations of spiking neurons for fine timing sound localization," in *Neural Networks (IJCNN), The 2013 International Joint Conference on*, pp. 1–8, Aug 2013.

[30] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[31] G. La Camera, M. Giugliano, W. Senn, and S. Fusi, "The response of cortical neurons to in vivo-like input current: theory and experiment," *Biological cybernetics*, vol. 99, no. 4-5, pp. 279–301, 2008.

[32] A. N. Burkitt, "A review of the integrate-and-fire neuron model: I. homogeneous synaptic input," *Biological cybernetics*, vol. 95, no. 1, pp. 1–19, 2006.

[33] A. J. Siegert, "On the first passage time probability problem," *Physical Review*, vol. 81, no. 4, p. 617, 1951.

[34] Q. Liu, "A gabor filter prefers the horizontal lines running on SpiNNaker in real-time ." `https://www.youtube.com/watch?v=PvJy6RKAJhw&feature=youtu.be&list=PLxZ1W-Upr3eoQuLxq87qpUL-CwSphtEBJ`, Sept. 2014.

[35] Q. Liu, "Feature extraction of live retinal input." `http://youtu.be/FZJshPCJ1pg?list=PLxZ1W-Upr3eoQuLxq87qpUL-CwSphtEBJ`, Sept. 2014.

[36] Q. Liu, "Live dynamic posture recognition on SpiNNaker." `http://youtu.be/yxN90aGGKvg?list=PLxZ1W-Upr3eoQuLxq87qpUL-CwSphtEBJ`, Sept. 2014.

[37] M. Elmezain, A. Al-Hamadi, J. Appenrodt, and B. Michaelis, "A hidden markov model-based isolated and meaningful hand gesture recognition," *International Journal of Electrical, Computer, and Systems Engineering*, vol. 3, no. 3, pp. 156–163, 2009.