BIOLOGICALLY-PLAUSIBLE OBJECT RECOGNITION USING SPIKING NEURONS

November 27, 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 What Is Object Recognition?

The definition of object recognition is well accepted [8] as the ability to assign labels to particular objects, ranging from precise labels ('identification') to course labels ('categorisation'). It involves the ability to accomplish the tasks under the various identity preserving transformations such as object position, scale, viewing angel, background clutter and etc.

The brain can accurately recognise and categorise objects remarkably quickly, e.g. the recognition time in monkeys takes less than 200 ms [9] and the images are presented sequentially in spikes less than 100 ms [10]. This research focuses on this rapid and highly accurate object recognition, ‘core recognition’, which is defined in [11].

1.2 Why Is It Important?

The brain recognises huge amount of objects rapidly and effortlessly even in cluttered and natural scenes. While the major stumbling crux of the computer object recognition systems lies in the invariance problem. Each encounter of an object on the retina is completely unique, because of the illumination (lighting conditions), position (projection locations on the retina), scale (distances and sizes), pose (viewing angles), and clutter (visual contexts) variabilities. In addition, a difficult specificity-invariance trade-off occurs in the categorisation tasks, since the recognition should be able to discriminate different object classes (intraclass variability) while at the same time tolerant to image transformations.

To explore the invariance problem in object recognition in a biologically plausible way is the right place to tackle the crux computational difficulty, since biological visual systems excel. Moreover, the energy-efficient manner will help in building object recognition systems, i.e. posture recognition, as human-machine interfaces in portable devices.

1.3 How to Mimic The Brain?

To explore how brain may recognise objects, we have employed a biologically-inspired DVS silicon retina [12], a good example of low-cost visual processing due to its event-driven and redundancy-reducing style of computation; and a SpiNNaker system [13], 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 different layers 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 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.

1.4 Report Structure

In Chapter 2, this report starts from the biology aspects of object recognition: the task is mainly processed along the ventral visual pathway with the untangling object representations at different level in the hierarchical abstractions. Followed by the introduction of spiking neural networks, the abilities of their single neurons and populations modelling are stated and the details of the hardware of the proposed neuromorphic system, including the silicon retina and the SpiNNaker platform are presented. In terms of the preliminary work, the neural network models are defined and tested on Matlab, and the model structures and experimental results are stated in Chapter 3. In Chapter 4, the rate-based models are converted into spiking neurons, and real-time live recognition and recorded data experiments are carried out. The contribution of this work is summarised and the future directions are provided in Chapter 5.

Chapter 2

Background

This chapter provides the readers with detailed biology background of object recognition in the brain and with the fundamental principles of neural modelling using spiking neural networks. This is followed by an introduction to the neuromorphic hardware platform specialised for neural simulations of visual processing.

2.1 How the Brain Represent Objects?

The central visual system consists of several cortical areas responsible for visual processing, which are placed in a hierarchical pattern according to the anatomical experiments [14]. There are two basic streams locating in the visual area: a dorsal and a ventral pathway (Figure [???]). They differ in behavioural patterns according to the observation from brain lesions [15], and also in functions where the dorsal pathway targets on the 'where' tasks and the ventral on the 'what'. The ventral ('what') visual stream holds the critical circuits for object recognition and stimulus identification, whereas the dorsal pathway ('where') pathway contributes to the processing of the spatial location of the stimulus [15, 16]. Another definition of the difference between these two pathways is a 'perception/action' dichotomy: the ventral stream, 'perception' pathway, cognises the world by means of object recognition and memory, while the dorsal stream, 'action' pathway, provides real-time visual guidance for motor actions such as eye movements and grasping objects [17].

This research mainly focuses on the ventral visual pathway, since it dominates the object recognition among the cortical areas. Thus, the dorsal pathway will beyond the scope of the this study.

2.1.1 The Ventral Visual Pathway

Decades of evidence argue that the primate ventral visual processing streama set of cortical areas arranged along the occipital and temporal lobes (Figure 3A)houses key circuits that underlie object recognition behavior. The ventral visual stream has been parsed into distinct visual areas based on anatomical connectivity patterns, distinctive anatomical structure, and retinotopic mapping (Felleman and Van Essen, 1991). Complete retinotopic maps have been re- vealed for most of the visual field (at least 40 degrees eccentricity from the fovea) for areas V1, V2, and V4 (Felleman and Van Essen, 1991) and thus each area can be thought of as conveying a population-based re-representation of each visually presented image. Within the IT complex, crude retinotopy exists over the more posterior portion (pIT; Boussaoud et al., 1991; Yasuda et al., 2010), but retinotopy is not reported in the central and anterior regions (Felleman and Van Essen, 1991). Thus, while IT is commonly parsed into subareas such as TEO and TE (Jans- sen et al., 2000; Saleem et al., 2000, 1993; Suzuki et al., 2000; Von Bonin and Bailey, 1947) or posterior IT (pIT), central IT (cIT), and anterior IT (aIT) (Felleman and Van Essen, 1991), it is unclear if IT cortex is more than one area, or how the term area should be applied. a hierarchical organization (as opposed to a parallel or fully interconnected organization) of the areas with visual information traveling first from the retina to the lateral geniculate nucleus of the thalamus (LGN), and then through cortical area V1 to V2 to V4 to IT (Felle- man and Van Essen, 1991). Consistent with this, the (mean) first visually evoked responses of each successive cortical area are successively lagged by about 10 ms (Nowak and Bullier, 1997; Schmolesky et al., 1998; see Figure 3B). Thus, just around 100 ms after image photons impinge on the retina, a first wave of image- selective neuronal activity is present throughout much of IT (e.g., Desimone et al., 1984; DiCarlo and Maunsell, 2000; Hung et al., 2005; Kobatake and Tanaka, 1994a; Logothetis and Shein- berg, 1996; Tanaka, 1996).

V1

V2

V4

IT- the main part. including IT single neuron behaviour.

2.1.2 Neural Code for Object Representation

we consider the neuronal representation in a given cortical area (e.g., the IT representation) to be the spatio-temporal pattern of spikes produced by the set of pyramidal neurons that project out of that area (e.g., the spiking patterns traveling along the population of axons that project out of IT; see Figure 3B). How is the spiking activity of individual neurons thought to encode visual information?

Single neuron representation

Most studies have investigated the response properties of neurons in the ventral pathway by assuming a firing rate (or, equivalently, a spike count) code, i.e., by counting how many spikes each neuron fires over several tens or hundreds of milli- seconds following the presentation of a visual image, adjusted for latency (e.g., see Figures 4A and 4B). Historically, this temporal window (here called the decoding window) was justi- fied by the observation that its resulting spike rate is typically well modulated by relevant parameters of the presented visual images (such as object identity, position, or size; Desimone et al., 1984; Kobatake and Tanaka, 1994b; Logothetis and Sheinberg, 1996; Tanaka, 1996) (see examples of IT neuronal responses in Figures 4A4C), analogous to the well-understood firing rate modulation in area V1 by low level stimulus properties such as bar orientation (reviewed by Lennie and Mov- shon, 2005).

population coding

Like all cortical neurons, neuronal spiking throughout the ventral pathway is variable in the ms-scale timing of spikes, re- sulting in rate variability for repeated presentations of a nominally identical visual stimulus. This spike timing variability is consistent with a Poisson-like stochastic spike generation process with an underlying rate determined by each particular image (e.g., Kara et al., 2000; McAdams and Maunsell, 1999). Despite this vari- ability, one can reliably infer what object, among a set of tested visual objects, was presented from the rates elicited across the IT population (e.g., Abbott et al., 1996; Aggelopoulos and Rolls, 2005; De Baene et al., 2007; Heller et al., 1995; Hung et al., 2005; Li et al., 2009; Op de Beeck et al., 2001; Rust and DiCarlo, 2010). It remains unknown whether the ms-scale spike variability found in the ventral pathway is noise (in that it does not directly help stimulus encoding/decoding) or if it is somehow synchronized over populations of neurons to convey useful, perhaps multi-plexed information (reviewed by Ermentrout et al., 2008).

50 ms window matters

IT neuronal spiking patterns (e.g., concatenated decoding windows, each less than 50 ms) does not convey significantly more information about object identity than larger time windows (e.g., a single, 200 ms decoding window), suggesting that the results of ventral stream processing are well described by a firing rate code where the relevant underlying time scale is 50 ms (Abbott et al., 1996; Aggelopoulos and Rolls, 2005; Heller et al., 1995; Hung et al., 2005). While different time epochs rela- tive to stimulus onset may encode different types of visual infor- mation (Brincat and Connor, 2006; Richmond and Optican, 1987; Sugase et al., 1999), very reliable object information is usually found in IT in the first 50 ms of neuronal response (i.e., 100150 ms after image onset, see Figure 4A). More specif- ically, (1) the population representation is already different for different objects in that window (DiCarlo and Maunsell, 2000), and (2) responses in that time window are more reliable because peak spike rates are typically higher than later windows (e.g., Hung et al., 2005).

2.1.3 IT

simple weighted summations of IT spike
Although visual information processing in the first stage of the ventral stream (V1) is reasonably well understood (see Lennie and Movshon, 2005 for review), processing in higher stages (e.g., V4, IT) remains poorly understood. Nevertheless, we know that the ventral stream produces an IT pattern of activity that can directly support robust, real-time visual object catego- rization and identification, even in the face of changes in object position and scale, limited clutter, and changes in background context (Hung et al., 2005; Li et al., 2009; Rust and DiCarlo, 2010). Specifically, simple weighted summations of IT spike counts over short time intervals (see section 2) lead to high rates of cross-validated performance for randomly selected popula- tions of only a few hundred neurons (Hung et al., 2005; Rust and DiCarlo, 2010) (Figure 4E), and a simple IT weighted sum- mation scheme is sufficient to explain a wide range of human invariant object recognition behavior (Majaj et al., 2012).

Better than lower level
Importantly, IT neuronal populations are demonstrably better at object identification and categorization than populations at earlier stages of the ventral pathway (Freiwald and Tsao, 2010; Hung et al., 2005; Li et al., 2009; Rust and DiCarlo, 2010).

A Contemporary View of IT Single Neurons
Respond to different variations
How do these IT neuronal population phenomena (above) depend on the responses of individual IT neurons? Under- standing IT single-unit responses has proven to be extremely challenging and while some progress has been made (Brincat and Connor,

2004; Yamane et al., 2008), we still have a poor ability to build encoding models that predict the responses of each IT neuron to new images (see Figure 4B). Nevertheless, we know that IT neurons are activated by at least moderately complex combinations of visual features (Brincat and Connor, 2004; Desimone et al., 1984; Kobatake and Tanaka, 1994b; Per- rett et al., 1982; Rust and DiCarlo, 2010; Tanaka, 1996) and that they are often able to maintain their relative object preference over small to moderate changes in object position and size (Brin- cat and Connor, 2004; Ito et al., 1995; Li et al., 2009; Rust and DiCarlo, 2010; Tove e et al., 1994), pose (Logothetis et al., 1994), illumination (Vogels and Biederman, 2002), and clutter (Li et al., 2009; Missal et al., 1999, 1997; Zoccolan et al., 2005).

respond to more objects

Contrary to popular depictions of IT neurons as narrowly selective object detectors, neurophysiological studies of IT are in near universal agreement with early accounts that describe a diversity of selectivity: We found that, as in other visual areas, most IT neurons respond to many different visual stimuli and, thus, cannot be narrowly tuned detectors for particular complex objects. (Desimone et al., 1984). For example, studies that involve probing the responses of IT cells with large and diverse stimulus sets show that, while some neurons appear highly selective for particular objects, they are the exception not the rule. Instead, most IT neurons are broadly tuned and the typical IT neuron responds to many different images and objects (Brincat and Connor, 2004; Freedman et al., 2006; Kreiman et al., 2006; Logothetis et al., 1995; Op de Beeck et al., 2001; Rolls, 2000; Rolls and Tovee, 1995; Vogels, 1999; Zoccolan et al., 2007; see Figure 4B).

conclusion

Such findings argue for a distributed representation of visual objects in IT, as suggested previously (e.g., Desimone et al., 1984; Kiani et al., 2007; Rolls and Tovee, 1995)a view that motivates the population decoding approaches described above (Hung et al., 2005; Li et al., 2009; Rust and DiCarlo, 2010). That is, single IT neurons do not appear to act as sparsely active, invariant detectors of specific objects, but, rather, as elements of a population that, as a whole, supports object recog- nition. This implies that individual neurons do not need to be invariant. Instead, the key single-unit property is called neuronal tolerance: the ability of each IT neuron to maintain its prefer- ences among objects, even if only over a limited transformation range (e.g., position changes; see Figure 4C; Li et al., 2009). Mathematically, tolerance amounts to separable single-unit response surfaces for object shape and other object variables such as position and size (Brincat and Connor, 2004; Ito et al., 1995; Li et al., 2009; Tove e et al., 1994; see Figure 4D). This contemporary view, that neuronal tolerance is the required and observed single-unit phenomenology, has also been shown for less intuitive identity-preserving transformations such as the addition of clutter (Li et al., 2009; Zoccolan et al., 2005).

Summery
Taken together, the neurophysiological evidence can be summarized as follows. First, spike counts in 50 ms IT decod- ing windows convey information about visual object identity. Second, this information is available in the IT population begin- ning 100 ms after image presentation (see Figure 4A). Third, the IT neuronal representation of a given object across changes in position, scale, and presence of limited clutter is untangled from the representations of other objects, and object identity can be easily decoded using simple weighted summation codes (see Figures 2B, 4D, and 4E). Fourth, these codes are readily observed in passively viewing subjects, and for objects that have not been explicitly trained (Hung et al., 2005). In sum, our view is that the output of the ventral stream is reflexively ex- pressed in neuronal firing rates across a short interval of time (50 ms) and is an explicit object representation (i.e., object identity is easily decodable), and the rapid production of this representation is consistent with a largely feedforward, nonlinear processing of the visual input.

2.1.4 Hierarchical Abstractions

Feed-forward, hierarchical organisation and abstraction.

We have arrived at a putative canonical meta job description, local subspace untangling, by working our way top-down from the overall goal of visual recognition and considering neuro- anatomical data. How might local subspace untangling be instantiated within neuronal circuits and single neurons? Historically, mechanistic insights into the computations per- formed by local cortical circuits have derived from bottom-upapproaches that aim to quantitatively describe the encoding functions that map image features to the firing rate responses of individual neurons. One example is the conceptual encoding models of Hubel and Wiesel (1962), which postulate the existence of two operations in V1 that produce the response properties of the simple and complex cells. First, V1 simple cells imple- ment AND-like operations on LGN inputs to produce a new form of selectivityan orientation-tuned response. Next, V1 complex cells implement a form of invariance by making OR- like combinations of simple cells tuned for the same orientation. These conceptual models are central to current encoding models of biological object recognition (e.g., Fukushima, 1980; Riesenhuber and Poggio, 1999b; Serre et al., 2007a), and they have been formalized into the linear-nonlinear (LN) class of en- coding models in which each neuron adds and subtract its inputs, followed by a static nonlinearity (e.g., a threshold) to produce a firing rate response (Adelson and Bergen, 1985; Carandini et al., 2005; Heeger et al., 1996; Rosenblatt, 1958). While LN-style models are far from a synaptic-level model of a cortical circuit, they are a potentially powerful level of abstraction in that they can account for a substantial amount of single-neuron response patterns in early visual (Carandini et al., 2005), somatosensory (DiCarlo et al., 1998), and auditory cortical areas (Theunissen et al., 2000). Indeed, a nearly complete accounting of early level neuronal response patterns can be achieved with extensions to the simple LN model frameworkmost notably, by divisive normalization schemes in which the output of each LN neuron is normalized (e.g., divided) by a weighted sum of a pool of nearby neurons (reviewed by Carandini and Heeger, 2011). Such schemes were used originally to capture luminance and contrast and other adaptation phenomena in the LGN and V1 (Mante et al., 2008; Rust and Movshon, 2005), and they represent a broad class of models, which we refer to here as the normalized LN model class (NLN; see Figure 5). We do not know whether the NLN class of encoding models can describe the local transfer function of any output neuron at any cortical locus (e.g., the transfer function from a V4 subpop- ulation to a single IT neuron). However, because the NLN model is successful at the first sensory processing stage, the parsimo- nious view is to assume that the NLN model class is sufficient but that the particular NLN model parameters (i.e., the filter weights, the normalization pool, and the specific static nonlinearity) of each neuron are uniquely elaborated. Indeed, the field has implicitly adopted this view with attempts to apply cascaded NLN-like models deeper into the ventral stream (e.g., David et al., 2006). Unfortunately, the approach requires exponentially more stimulus-response data to try to constrain an exponentially expanding set of possible cascaded NLN models, and thus we cannot yet distinguish between a principled inadequacy of the cascaded NLN model class and a failure to obtain enough data. This is currently a severe in practice inadequacy of the cascaded NLN model class in that its effective explanatory power does not extend far beyond V1 (Carandini et al., 2005). Indeed, the problem of directly determining the specific image- based encoding function (e.g., a particular deep stack of NLN models) that predicts the response of any given IT neuron (e.g., the one at the end of my electrode today) may be practically impossible with current methods.

2.2 Spiking Neural Network

2.2.1 Neuronal Model

The Membrane Potential

A typical neuron is divided into three parts: the dendrites, the soma and the axon. Generally speaking, the dendrites receive the input signals from the previous neurons. The soma is where the received input signals are being processed and the axon is where the output signals are transmitted. The synapse is between every two neurons; if a neuron j sends a signal across the synapse to neuron i, the neuron that sends the signal is called presynaptic and the neuron that receives the signal is called postsynaptic neuron. Hodgkin and Huxley [**?**] found out, by experimenting on the squid giant axon, that it is the time of the spikes that encodes information [**?**], Figure 2.1.

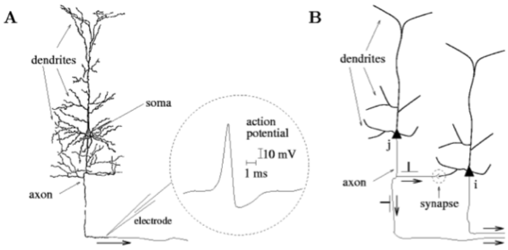


Figure 2.1: A. The inset shows an example of a neuronal action potential. The action potential is a short voltage pulse of 1-2ms duration and 100mV of amplituted. B. Signal transmition from a presynaptic neuron j to a post synaptic neuron i. The synapse is marked by a dashed circle [**?**].

A living neuron maintains a voltage drop across its membrane. Every cell is surrounded by positive and negative ions. The main ions are K$^+$ (Potassium), Cl$^-$ (chloride), Na$^+$ (sodium) and Ca^{2+} (calcium). In the inner surface of the membrane there is an excess of negative charges and on the outer surface there is an excess of positive charges. Those charges create the membrane potential.

The membrane potential can be calculated from the following equation: Vm=Vin-Vout, where Vin is the negative charges on the inside of the cell and Vout are the positive charges outside of the cell. When the membrane potential is at the resting state, that is when it is not receiving any input signals, the resting potential Vrest is set to Vin, which is around -60mV to -70mV.

When the neuron receives an input, some of the ion channels of the cell open and others close, resulting in an electrical current flow into the cell, which results in a change of the resting potential Vrest [?].

The phenomenon during which the membrane's potential changes exceed the resting potential is called depolarization. The opposite phenomenon is called hyperpolarization. When the depolarization reaches a critical value, also known as threshold, the cell produces an action potential (a spike) [?], figure 1.2. If the membrane potential receives an input that causes depolarization or hyperpolarization and after that does not receive any other input, the membrane potential returns slowly to its resting potential.

In the case of the Glial cell the potassium K+ are flowing from the inside of the cell to the outside causing a potential difference called equilibrium potential Ek [?] . This E_k determines the resting membrane potential and can be calculated from the Nerst Equation:

$$E_k = \frac{RT}{zF} ln \frac{[X]_o}{[X]_i} \tag{2.1}$$

Where R is the gas constant, T is the temperature in Kelvin, z is the valence of the ion, F the Faraday constant, $[X]_o$ and $[X]_i$ are the concentrations of the ion outside and inside of the cell [?]. Thus the Vrest for the Glial cell is Vrest = -75mV. The membrane potential will be discussed in the next section when the Hodgkin-Huxley neuron model will be described based on the experiments on the squid giant axon.

The Action Potential

As stated before, when the membrane potential reaches a critical value called threshold it emits an action potential, also known as a spike. This is caused by the movement of ions across the membrane through voltage-gated channels [?]. The spikes are identical to each other and their form does not change as the signal moves from a presynaptic to a postsynaptic neuron [?]. The firing times of a neuron are called spike train and it is represented with the following equation:

$$Fi = \{t_i^1, t_i^2, ..., t_i^n\} \tag{2.2}$$

The subscript i defines the neuron and the superscript defines the number of the emitted spikes, where n is the most recent emitted spike.

Directly after the transmission of a spike, the membrane potential goes through a phase of high hyperpolarization under the resting potential and then slowly returns back to the resting potential. During that time, it is not possible to emit a second spike even for strong input signals, that is because the ion channels are open instantly after a spike has been generated [?]. The minimum time between two generated spikes is called absolute refractory period and the phenomenon where the membrane potential undershoots below the resting potential is known as the spike after potential (SAP), Figure 2.2.

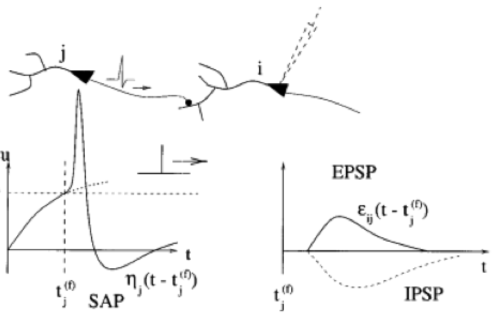


Figure 2.2: The membrane potential is increased and at time tj(f) the membrane potential reaches the threshold so a spike is emmited [?].

The Synapse

Between the axon of the presynaptic neuron and the dendrite of the postsynaptic neuron there is a small gap, also known as synaptic gap. The operation of the synapse is very complicated and a detailed description is beyond the scope of this review. The spike of the presynaptic neuron cannot cross this gap, however, when a spike arrives from the presynaptic neuron to the synapse the gap is filled a fluid that generates a postsynaptic potential (PSP) to the dendrite of the postsynaptic neuron [?]. This process does not happen instantaneous; there is a small delay generated in that particular synapse.

There are two types of postsynaptic potentials. If the generated postsynaptic potential is positive it is called excitatory postsynaptic potential (EPSP) or if the generated postsynaptic potential is negative it is call inhibitory postsynaptic potential (IPSP), Figure 2.3. An IPSP lowers the membrane potential of the postsynaptic neuron while an EPSP increases it and may cause it to fire a spike.

Spiking Neuron Models

Spiking neuron models can be divided into two major categories [?] based on their level of abstraction: The conductance models and the threshold models. The conductance models simulate the ion channels of the cell, while the threshold models represent a higher level of abstraction where the threshold voltage has a fixed value and the neuron fires every time the membrane potential reaches it.

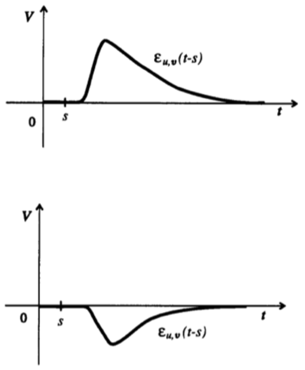


Figure 2.3: Excitatory postsynaptic potential (EPSP) and Inhibitory postsynaptic potential (IPSP) of a biological neuron [?].

There are two additional models that will not be described in this thesis: the compartmental and rate models. The compartmental models will not be discussed due to their complexity and the rate models are actually the sigmoidal neurons that are used in the traditional artificial neural networks of the 2nd generation. Due to their nature, they neglect all the temporal information of the spikes and only describe their activity as spike rates.

In general, Conductance-Based models have been derived from the Nobel prize winners (1963) Hodgkin and Huxley, based on the experiments that they performed on the giant axon squid [?]. Basically, they describe what happens to the ion channels of the neuron cell.

Leaky-Integrate-and-Fire Model

The Leaky Integrate-and-Fire neurons are threshold-fire models that are based on the summation of all contributions of the presynaptic neurons to the membrane potential. If the membrane potential reaches a fixed threshold from below, the neuron will fire and after an axonal delay it will cause neurotransmitter release from the synapses.

They have been extensively used in large spiking neural networks [?] because of their ease of implementation and the low computational cost.

The basic circuit of the integrate-and-fire model can be seen in Figure 2.4. It consists of a resistor R in parallel with a capacitor C that models the passive patch of the membrane. In addition, a reset mechanism has been added, as a switch that closes when the membrance potential reaches a threshold value from below.

Using the Ohm's law, the schematic in the Figure 2.4 can be described by the following equation:

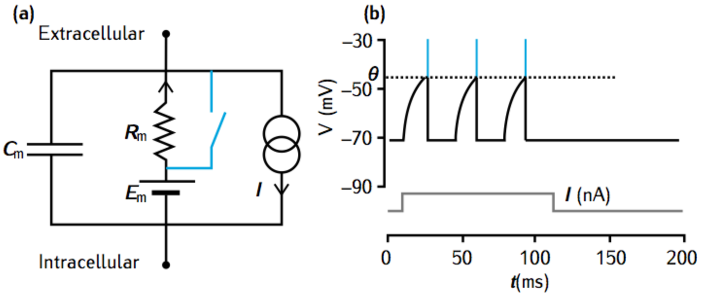


Figure 2.4: The Leaky Integrate-and-fire model. (a) The RC circuit diagram of the model. When the membrane potential reaches a threshold voltage θ, the neuron is considered to have fired a spike and the switch closes. The aforementioned short-circuit causes the membrane potential to return back to the resting membrane potential E$_m$. (b) Response of LIF circuit to current injection. The refractory period can be observed directly after the firing of a spike [**?**].

$$C_m \frac{dV}{dt} = -\frac{V - E_m}{R_m} + I \tag{2.3}$$

Where C$_m$ is the membrance capacitance, R$_m$ is the membrane reistance and I is the total current flowing into the cell, which could be from an electrode or from other synapses, see equation 2.5. Furthermore, by setting τ_m = RC, the equation 2.3 could be rewritten as:

$$\tau_m \frac{dV}{dt} = -V + E_m + R_m I \tag{2.4}$$

Where τ_m is known as membrane time constant. When the membrane potential reaches a predefined threshold θ, the neuron fires a spike and the membrane potential is reset to E$_m$.

The electrical current resulting from the neurotransmitter at time t$_s$ is, for t ¿= t$_s$:

$$I_{syn}(t) = g_{syn}(t)(V(t) - E_{syn}) \tag{2.5}$$

Where g$_{syn}$(t) is the synaptic conductance which peaks at $\bar{g}_{syn}$, V(t) is the membrane voltage of the postsynaptic neuron, for conductance-based synapses (COBA) and E$_{syn}$ is the reversal potential of the synaptic conductance. In many cases equation 2.5 can be simplified so that synapses can be thought of as sources of current instead of conductance (CUBA), this is done by setting V = V$_{rest}$ in equation 2.5. This simplification is a good approximation for excitatory synapses but not for inhibitory synapses where the inhibitory reversal potential could be close or even above the resting potential [?]. The three commonly used equations for the synaptic conductance are the: (a) single exponential decay, (b) alpha function and (c) dual exponential function:

$$g_{syn} = \bar{g}_{syn} exp(-\frac{t-t_s}{\tau}) \tag{2.6a}$$

$$g_{syn} = \bar{g}_{syn} \frac{t-t_s}{\tau} exp(-\frac{t-t_s}{\tau}) \tag{2.6b}$$

$$g_{syn} = \bar{g}_{syn} \frac{\tau_1 \tau_2}{\tau_1 - \tau_2} (exp(-\frac{t-t_s}{\tau_1}) - exp(-\frac{t-t_s}{\tau_2}) \tag{2.6c}$$

Finally, a number of variations of the Leaky Integrate-and-Fire have been proposed to model more complex firing patterns such as the firing rate adaptation or bursting (Type II firing). These models are the Quadratic Integrate-and-Fire model and the Exponential Integrate-and-Fire-model [?].

2.2.2 Spike Coding

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) [18]. Spatio-temporal information can also be stored in the exact firing time rather than relative delays [19]. Maass [20] 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.

2.2.3 Rate Coding

In rate coding the information is encoded into the mean firing rate of the neuron also known as temporal average [?]:

$$v = \frac{n_{sp}(T)}{T} \tag{2.7}$$

Where T is time window, nsp(T) are the number spikes emitted during the time window. There are three averaging procedures [?]: Rate as a spike count (average over time), rate as a spike density (average over several runs) and rate as a population activity (average over several neurons).

2.2.4 Temporal Coding

In temporal coding the information is encoded in the form of spike times [?]. Hopfield [?] has proposed a method for encoding analogue data into timing of the spikes with respect to an oscillatory pattern of activity. This method has been proven experimentally in the electric fish. In addition, Maass [?] proposed a method of encoding analogue information in the form of firing times. A different approach has been suggested by Wen and Sendhoff [?], where the input neurons encode information directly into spiking times and an additional bias neuron is used as a time reference. Finally, in polychronization [?], proposed by Izhikevich, the synaptic delays are tuned so that a neuron would respond to particular spatio-temporal patterns of activity.

2.2.5 Population Coding

In population coding a number of input neurons (population) are involved in the analogue encoding and produce different firing times. Bohte et al. [?] proposed a way of representing analogue input values into spike times using population coding. Multiple Gaussian Receptive Fields (GRF) were used so that the input neurons will encode an input value into spike times, Figure 2.5.

Firstly the range of the input data has to be calculated. Then the values Imax and Imin, which are the maximum and minimum values of the input data, have to be defined. Furthermore, the number of GRF neurons that are going to be used has to be chosen through the m variable. Lastly, the center of each GRF neuron is calculated from C$_i$ while the width of each GRF neuron is calculated by σ_i [?]:

$$C_i = I_{min} + \frac{(2i-3)}{2}\frac{(I_{max}-I_{min})}{m-2} \tag{2.8a}$$

$$\sigma_i = \frac{1}{\gamma}\frac{(I_{max}-I_{min})}{m-2} \tag{2.8b}$$

Where γ is constant number usually around 1.5. A threshold value has to be used so that GRF neurons, that are below the threshold, should not fire. In the example of

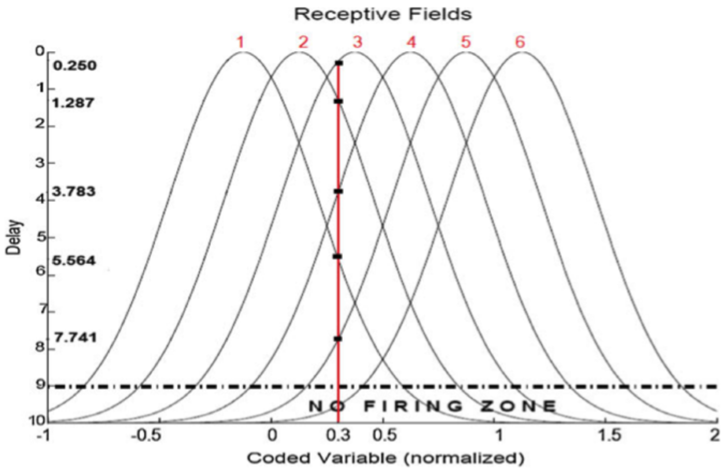


Figure 2.5: Encoding with Gaussian Receptive Fields. The horizontal axis represents the real input data, the vertical axis represent the firing times of the input neurons to an input value 0.3 [?].

Figure 2.5 the analogue value 0.3 is encoded into firing times of neuron 3 (0.250ms), neuron 2 (1.287ms), neuron 4 (3.783ms), neuron 1 (5.564ms) and neuron 5 (7.741ms). Neuron 6 does not emit a spike because it's below the threshold.

A different approach to population encoding was proposed by Eliasmith et al. in [?], where the firing rates of a heterogeneous population of neurons is used to encode an analogue value. Also, the rank-order coding proposed in [?] encodes an input value based on the order of spikes of a population.

2.2.6 Learning

In 1949 Hebb formulated the famous Hebb law [?]: ”When an axon of cell A is near enough to excite cell B or repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased”.

Hebb's law is modified so that the weights are updated based on the pre and postsynaptic activity of the neurons, also known as Spike Time Dependent Plasticity (STDP) [?].

In Figure 2.6 neuron j is the presynaptic neuron, neuron i is the postsynaptic neuron and t_j^f is the presynaptic fire time and tif is the postsynaptic fire time.

Furthermore, Bi and Poo [?, ?] found out that the synaptic efficacy Δw_{ij} is a function of the spike times of the presynaptic and postsynaptic neurons. Hence the term Spike Timing-Dependent Plasticity.

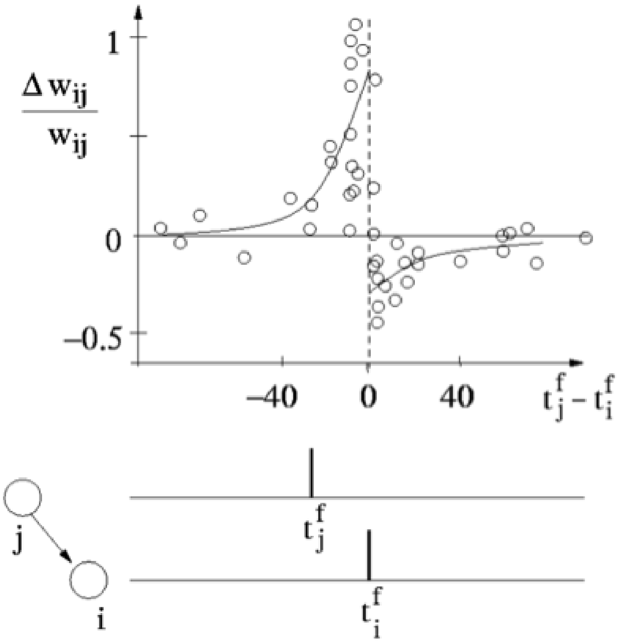


Figure 2.6: The weights are changing only if the firing times of neurons j and i are close to each other. Data taken from the experiments of Bi and Poo [**?**].

A way to calculate the synaptic weight updates has been proposed by Gerstner et al. [**?**] with the use of exponential learning windows:

$$\Delta W = \begin{cases} A_{+}exp(s/\tau_1) & \text{if } s \text{ ¡ } 0 \\ A_{-}exp(s/\tau_2) & \text{if } s \text{ ¿ } 0 \end{cases} \tag{2.9}$$

Where s = $t_j^{(f)}$-$t_i^{(f)}$ is the time difference between presynaptic and postsynaptic firing times. The τ_1 and τ_2 are constants and the A_{+} and A_{-} are used for stability issues in order to cap the weights to a maximum and minimum value, Figure 2.7.

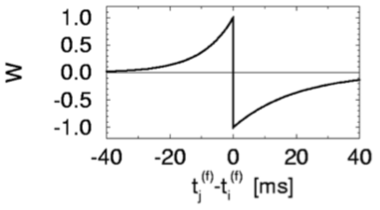


Figure 2.7: The exponential learning window as a function the difference between the presynaptic and the postsynaptic firing times. A_{+}=1, A_{-}=-1, τ_1=10ms, τ_2=20ms [**?**].

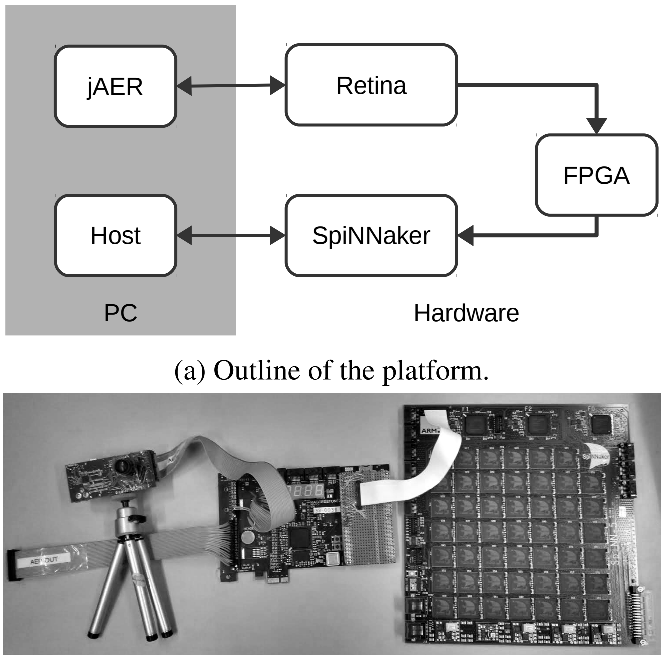
Numerous methods have been proposed in order to overcome the need of capping the weights to maximum and minimum values for unsupervised learning. For example, the BCM theory, named after the names of the authors (**B**ienenstock, **C**ooper, **M**unro) and it is based on experiments they conducted on neurons in the primary sensory cortex [**?**]. This method uses a sliding threshold mechanism to overcome the saturation of the weights during STDP. However, this method is too computationally expensive thus making it inadequate for large-scale simulations.

2.2.7 Successful Applications

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 [21]. Lee et al. [22] have implemented direction selective filters in real time using spiking neurons, considered as a convolution layer in the model of a so called CNN [23]. 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 [24] 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 [25]. Nengo [26] 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 [27]. An FPGA implementation of a Nengo model for digit recognition has been reported [28]. Deep Belief Networks (DBNs), the 4th generation of artificial neural network, have shown great success in solving classification problems. Recent study [29] in this area has mapped an offline-trained DBN onto an efficient event-driven spiking neural network for digit recognition tasks with resounding success.

2.3 Platforms

The outline of the platform is illustrated in Figure 2.8a, where the hardware system is configured, controlled and monitored by the PC. The jAER [30] 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 [31] the spiking activity of the network in real-time. The photograph of the



(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 2.8: System overview of the dynamic hand posture recognition platform.

hardware platform, Figure 2.8b, shows that the silicon retina connects to the SpiNNaker 48-node system via a Spartan-6 FPGA board [32].

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 [33]) 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 [12].

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 2.9 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 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 **??**. 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 [34].

From a neural modelling point of view, interfacing the silicon retina is performed using pyNN [35]. 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[32]. Besides the retina, we have successfully connected an AER based silicon cochlea [36] to SpiNNaker for a sound localisation task [37].

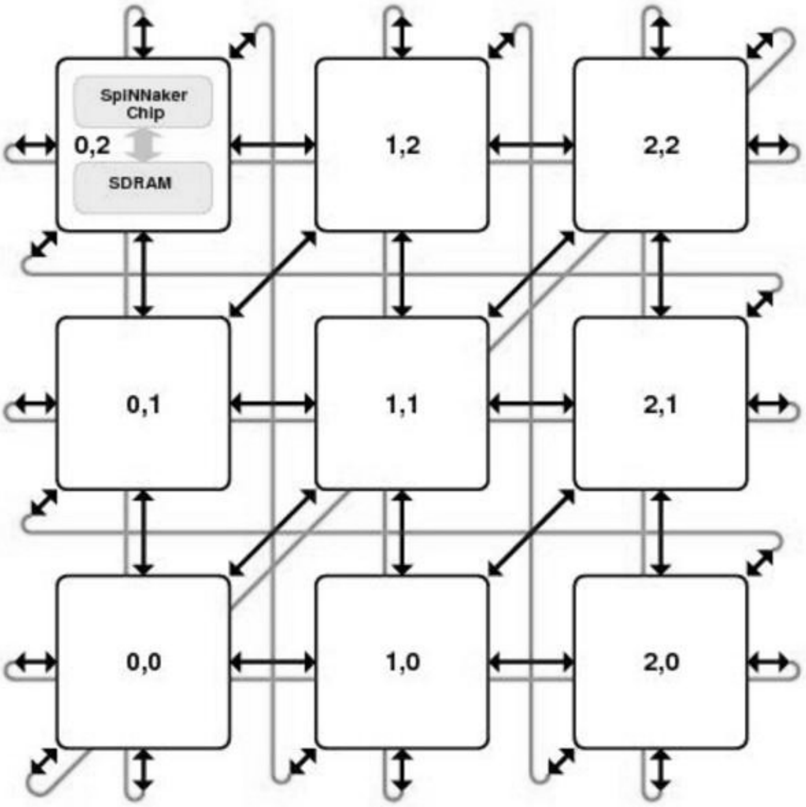


Figure 2.9: SpiNNaker system diagram. Each element represents one chip with local memory. Every chip connects to its neighbours through the six bi-directional on-board links.

Chapter 3

Convolutional Neural Networks

The convolutional network is well-known as an example of a biologically-inspired model. Figure 3.1 shows a typical convolutional connection between two layers of neurons. The repeated convolutional kernels are overlapped in the receptive fields of the input neurons.

3.1 Model Description

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 3.2 the first layer is the retina input,

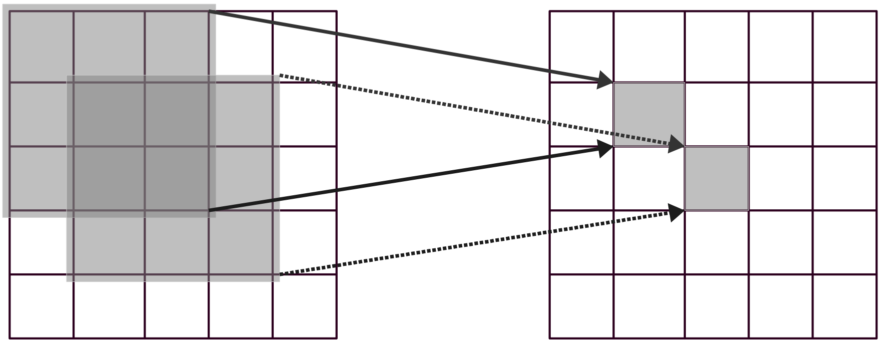


Figure 3.1: Each individual neuron in the convolution layer (right matrix) connects to its receptive field using the same kernel. The value of the kernel is represented by the synaptic weights between the connected neurons.

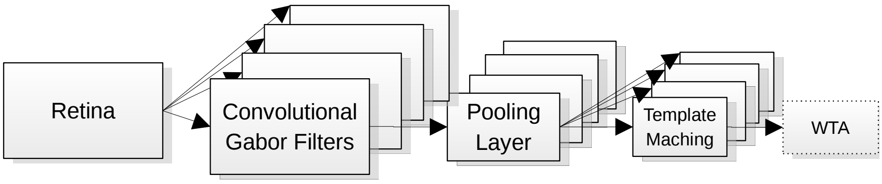


Figure 3.2: Model 1. The retina input is convolved with Gabor filters in the second layer, and then shrinks the sizes in the pooling layer. The templates are considered as convolution kernels in the last layer. The WTA circuit can be used as an option to show the template matching result more clearly.

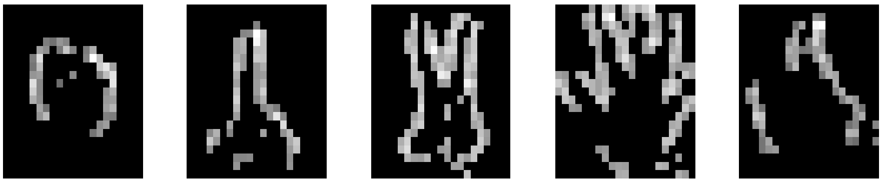


Figure 3.3: Templates of the five postures: 'Fist','Index Finger', 'Victory Sign', 'Full Hand' and 'Thumb up'.

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 3.3.

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

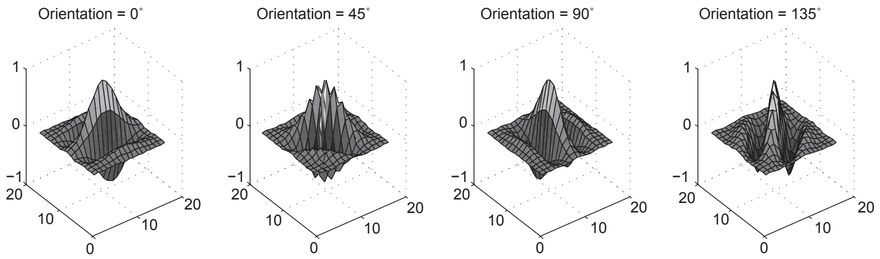


Figure 3.4: Real parts of the Gabor filters orienting four directions.

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 3.4) 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 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 [38], we designed a combined network model with MLP and CNN (Figure 3.5). 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

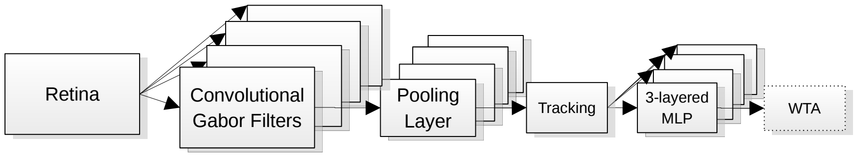


Figure 3.5: Model 2. The retina input convolves with Gabor filters in the second layer, and then shrinks the sizes in the pooling layer. The following tracking layer finds the most active area of some fixed size, moves the posture to the centre and pushes the image to the trained MLP. The winner-take-all (WTA) layer can be used as an option to show the template matching result more clearly.

plays an important role to find the most active region and forward the centred image to the next layer.

3.2 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.

3.3 Experimental Results

In Figure 3.6 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 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

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

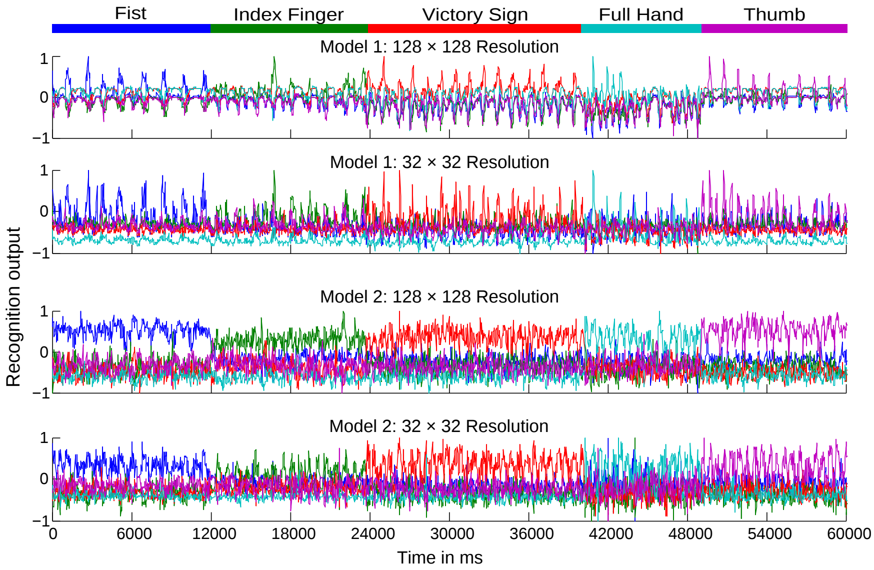


Figure 3.6: Neural responses with time of four experiments to the same recorded moving postures. The recognition output is normalised to [-1, 1]. Every point represents the highest response in a specific population (different colour) for a 30 ms frame. The 1st plot refers to Model 1 with the full input 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.

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 3.6 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 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.

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

Chapter 4

Recognition on SpiNNaker

4.1 Moving from Perceptrons to Spiking Neurons

It remains a challenge to transform traditional artificial neural networks into spiking ones. There are attempts [39] [40] 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

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

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 [41],

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

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.

4.2 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.

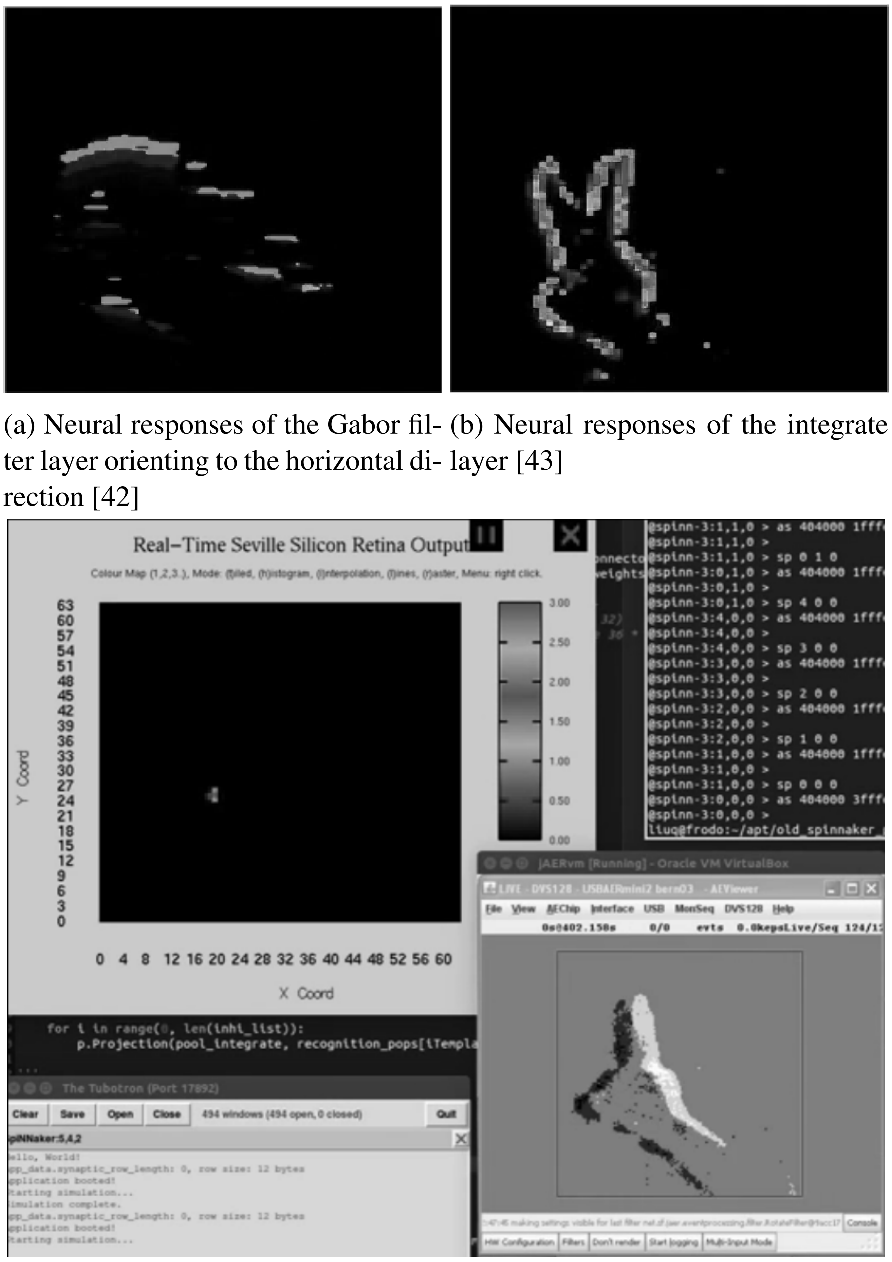
Figure 4.1 shows snapshots of neural responses of some populations during real-time recognition. Figure 4.1a is a snapshot of the Gabor population which prefers the horizontal direction, given the input posture of a 'Fist'; and Figure 4.1b shows the activity of the neurons in the integration layer, given 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 [42, 43, 44].

4.3 Recognition of Recorded Data

To compare with the results of the experiments carried out with Matlab (in Section 3.3), 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



(a) Neural responses of the Gabor filter layer orienting to the horizontal direction [42]

(b) Neural responses of the integrate layer [43]

```
@spinn-3:1,1,0 > as 404000 1fff
@spinn-3:1,1,0 >
@spinn-3:1,1,0 > sp 0 1 0
@spinn-3:0,1,0 > as 404000 1fff
@spinn-3:0,1,0 >
@spinn-3:0,1,0 > sp 4 0 0
@spinn-3:4,0,0 > as 404000 1fff
@spinn-3:4,0,0 >
@spinn-3:4,0,0 > sp 3 0 0
@spinn-3:3,0,0 > as 404000 1fff
@spinn-3:3,0,0 >
@spinn-3:3,0,0 > sp 2 0 0
@spinn-3:2,0,0 > as 404000 1fff
@spinn-3:2,0,0 >
@spinn-3:2,0,0 > sp 1 0 0
@spinn-3:1,0,0 > as 404000 1fff
@spinn-3:1,0,0 >
@spinn-3:1,0,0 > sp 0 0 0
@spinn-3:0,0,0 > as 404000 3fff
@spinn-3:0,0,0 >
liug@frodo:~/apt/old_spinnaker_
```

```
    for i in range(0, len(inhi_list)):
        p.Projection(pool_integrate, recognition_pops[iTempla
...
```

```
SpiNNaker:5,4,2
ello, World!
pp_data.synaptic_row_length: 0, row size: 12 bytes
pplication booted!
tarting simulation...
imulation complete.
pp_data.synaptic_row_length: 0, row size: 12 bytes
pplication booted!
tarting simulation...
```

```
0s@402.150s    0/0    evts  0.0kepsLive/Seq 124/1
```

```
:47:45 making settings visible for last filter net.sf.jaer.eventprocessing.filter.RotateFilter@1acc17
```

Real−Time Seville Silicon Retina Output
Colour Map (1,2,3..), Mode: (t)iled, (h)istogram, (i)nterpolation, (l)ines, (r)aster, Menu: right click.
Y Coord
63 60 57 54 51 48 45 42 39 36 33 30 27 24 21 18 15 12 9 6 3 0
X Coord
0 4 8 12 16 20 24 28 32 36 40 44 48 52 56 60
3.00 2.50 2.00 1.50 1.00 0.50 0.00
onnecto
weights
The Tubotron (Port 17892)
Clear
Save
Open
Close
494 windows (494 open, 0 closed)
Quit
JAERvm [Running] - Oracle VM VirtualBox
LIVE - DVS128 - USBAERmini2 bern03 - AEViewer
File View AEChip Interface USB MonSeq DVS128 Help
Console
HW Configuration
Filters
Don't render
Start logging
Multi-Input Mode

(c) Snapshot of the neuron responses of the template matching layer [44]

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

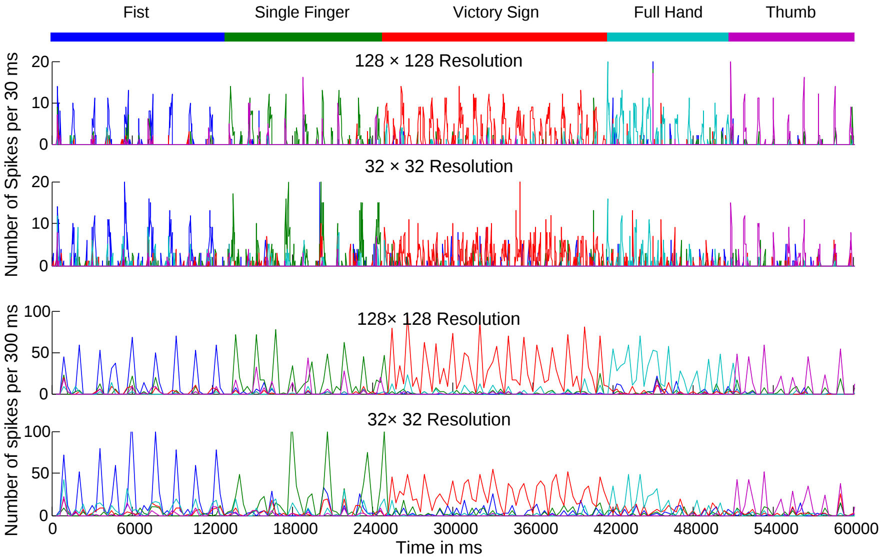
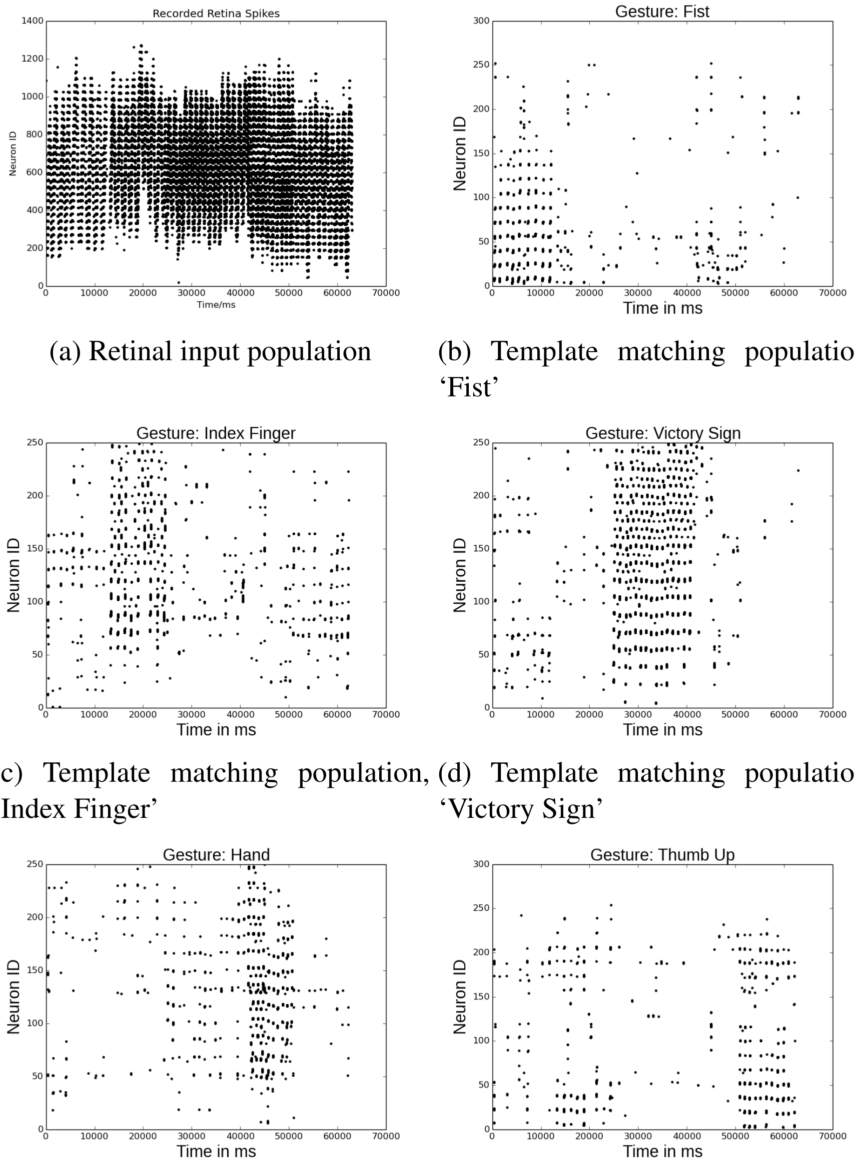


Figure 4.2: Real-time neural responses of two experiments on SpiNNaker with time to the same recorded postures. These two experiments only differ in input resolution. The result of the high input resolution test is plotted the first with a sample frame of 30 ms; while the 3rd plot shows the same result 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

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



(a) Retinal input population

(b) Template matching population, 'Fist'

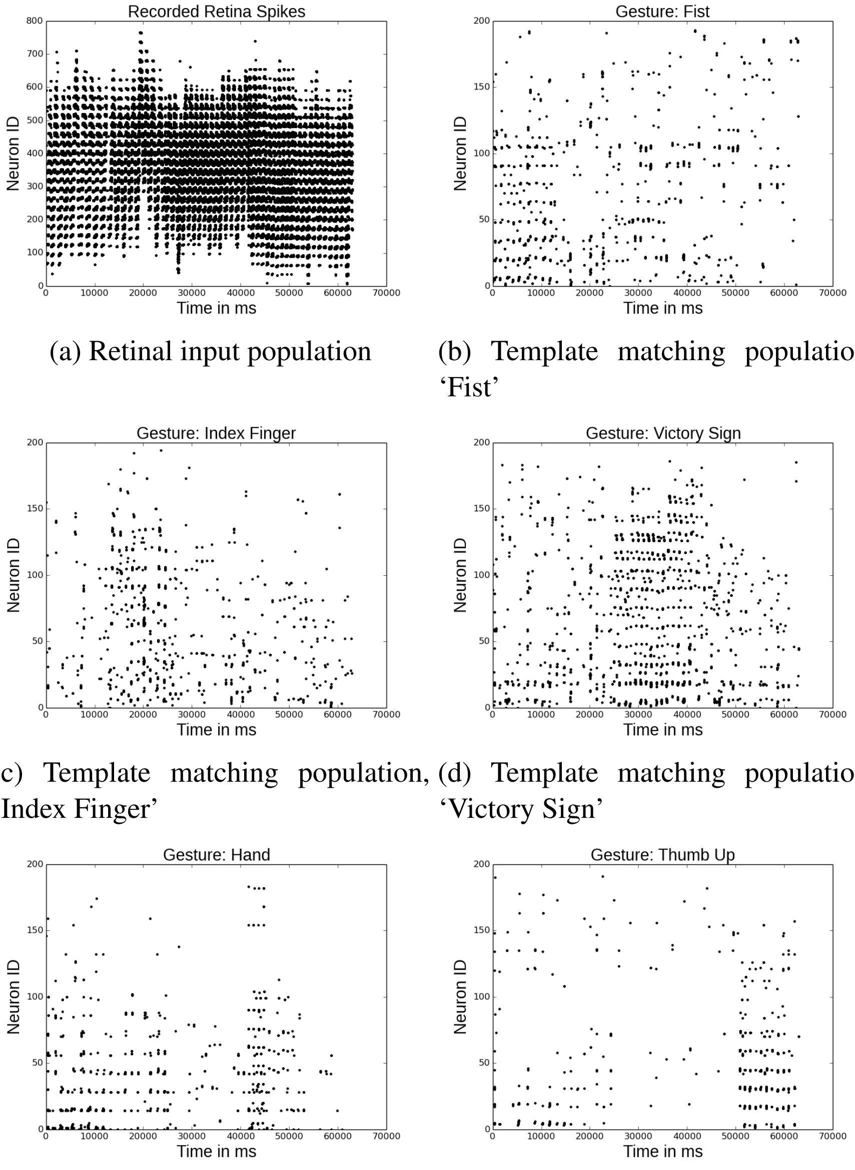
(c) Template matching population, 'Index Finger'

(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

Contributions and Research Plan

5.1 Contributions

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.

5.2 Future Work

5.2.1 Modelling The Ventral Visual Pathway with Spiking Neurons

5.2.2 Learning Between the Hierarchy Layers

5.2.3 Comparing with Biological Data

5.2.4 Building Dataset

5.2.5 Optional: Action Recognition

vision attention. short-term memory.

5.2.6 Optional: Sensor Fusion with Auditory Processing

platform. applications as lip-reading and speaker identification.

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. J. DiCarlo, D. Zoccolan, and N. C. Rust, "How does the brain solve visual object recognition?," *Neuron*, vol. 73, no. 3, pp. 415–434, 2012.

[9] M. Fabre-Thorpe, G. Richard, and S. J. Thorpe, "Rapid categorization of natural images by rhesus monkeys," *Neuroreport*, vol. 9, no. 2, pp. 303–308, 1998.

[10] C. Keysers, D.-K. Xiao, P. Földiák, and D. Perrett, "The speed of sight," *Journal of cognitive neuroscience*, vol. 13, no. 1, pp. 90–101, 2001.

[11] J. J. DiCarlo and D. D. Cox, "Untangling invariant object recognition," *Trends in cognitive sciences*, vol. 11, no. 8, pp. 333–341, 2007.

[12] 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.

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

[14] D. J. Felleman and D. C. Van Essen, "Distributed hierarchical processing in the primate cerebral cortex," *Cerebral cortex*, vol. 1, no. 1, pp. 1–47, 1991.

[15] J. Prado, S. Clavagnier, H. Otzenberger, C. Scheiber, H. Kennedy, and M.-T. Perenin, "Two cortical systems for reaching in central and peripheral vision," *Neuron*, vol. 48, no. 5, pp. 849–858, 2005.

[16] L. G. Ungerleider and J. V. Haxby, "what and where in the human brain," *Current Opinion in Neurobiology*, vol. 4, no. 2, pp. 157 – 165, 1994.

[17] M. A. Goodale and A. D. Milner, "Separate visual pathways for perception and action," *Trends in neurosciences*, vol. 15, no. 1, pp. 20–25, 1992.

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

[19] 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.

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

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

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

[42] 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.

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

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