

Neuromorphic Technologies for Defence and Security

Paul Kirkland^a, Gaetano Di Caterina^a, John Soraghan^a, and George Matich^b

^aNeuromorphic Sensing and Processing Lab, Electronic and Electrical Engineering Department, University of Strathclyde, Glasgow, UK

^bLeonardo MW, Basildon, UK

ABSTRACT

Despite the highly promising advances in Machine Learning (ML) and Deep Learning (DL) in recent years, DL requires significant hardware acceleration to be effective, as it is rather computationally expensive. Moreover, miniaturisation of electronic devices requires small form-factor processing units, with reduced SWaP (Size, Weight and Power) profile. Therefore, a completely new processing paradigm is needed to address both issues. In this context, the concept of neuromorphic (NM) engineering provides an attractive alternative, seen as the analog/digital implementation of biologically brain inspired neural networks. NM systems propagate spikes as means of processing data, with the information being encoded in the timing and rate of spikes generated by each neuron of a so-called spiking neural network (SNN). Based on this, the key advantages of SNNs are: less computational power required, more efficient and faster processing, much lower power consumption. This paper reports on the current state of the art in the field of NM systems, and it describes three application scenarios of SNN-based processing for security and defence, namely target detection and tracking, semantic segmentation, and control.

Keywords: neuromorphic, deep learning, machine learning, SNN

1. INTRODUCTION

In the last decade there have been significant developments in the field of AI, defined as any simulation of human intelligence exhibited by machines. More specifically, this process was driven by advances in techniques and technologies in the AI subfields of Machine Learning (ML) and Deep Learning (DL), which aim to teach computers the ability to perform tasks with data, without explicit programming.^{1,2} ML is a branch of AI and it uses numerical and statistical approaches to encode learning into mathematical models, which are then used to infer information from new data never seen before. Instead DL is a technique for implementing ML, and it stems from the work on Artificial Neural Networks (ANN). However DL is different in the sense that it uses deeper and more complex structures, generally referred to as Deep Neural Networks (DNN), as compared to traditional ANNs.²⁻⁴

The ground work for such advances was initiated in the academic research communities in 1980's, and only in early 2000's it came to attention of a wider audience, thanks to the widespread of the Internet and the rise of the modern big tech corporations, which embraced ML and DL technologies quite rapidly. In more recent years, a wide range of industries have recognised the potentials made available by AI in their respective fields of application, and so now almost any tech-related company is keen to invest and exploit the benefits of AI. In turn, this has convinced governments to invest a considerable level of funding in AI, through various initiatives. For example, The UK Government announced in the Spring Budget 2017 that the Industrial Strategy Challenge Fund would support initial investment of £270m, towards the development of disruptive technologies, including cutting-edge AI.⁵⁻⁷ The EU increased its annual investments in AI by 70%, as part of the Horizon 2020 programme, with the aim of reaching €1.5bn for the period 2018-2020. And this commitment is expected to increase further under the next framework Horizon Europe.⁸

Further author information: (Send correspondence to Gaetano Di Caterina.)

Gaetano Di Caterina: E-mail: gaetano.di-caterina@strath.ac.uk

Paul Kirkland: E-mail: paul.kirkland@strath.ac.uk

John J. Soraghan: Email: j.soraghan@strath.ac.uk

DL and DNNs have demonstrated to be very effective in many contexts and applications. However, these structures are rather complex and redundant. Indeed it is a well known fact that DL is computationally expensive, and it simply would not be affordable without all the dedicated processing units, hardware architectures and accelerators that have been developed in the last decade. In fact, a clear game changer in the effective application of DL was the use of GPUs on graphics card, to drastically reduce the network training time, by exploiting the massive parallel power of these processors. In terms of computational complexity, a clear bottleneck in ANNs and DNNs is represented by the backpropagation algorithm, which does give good results, but it is time consuming and can lead to the ‘vanishing gradient’ problem. This issue is only exacerbated by the very large volume of data that is generally required to train DL systems. As a consequence, a high amount of power is still required to train and run a DNN. All these issues pose a considerable challenge in the deployment of DL architectures on embedded and portable devices, and more in general in all those applications that require a reduced SWaP (Size, Weight and Power) profile. Therefore, a completely new processing paradigm is needed to address both issues.

In this context, the concept of neuromorphic (NM) engineering provides an attractive alternative, and it refers to analog, or combined analog and digital, implementation of biologically brain-inspired sensing and processing systems. The key characteristic of NM systems is that they generate and propagate spikes as means of processing data.⁹ Therefore NM represents a completely different computational paradigm, when compared to traditional Von Neumann-based processing. In general, NM include biologically brain-inspired sensors, processors, and algorithms. But in all three of these components, the key advantages of NM are: less computational power required, more efficient and faster processing, much lower power consumption. Spiking Neural Networks (SNN) are a particular type of neural networks where the information is indeed encoded in the timing and rate of spikes, which are generated by each neuron in the network once a set of incoming stimuli crosses an inner threshold within the neuron.¹⁰ Due to their spiking nature, SNNs are algorithms best suited for implementation in NM systems.

The NM concept dates back to the 1980’s, when researchers tried to morph the human retina,¹¹ and it has evolved since then. Nonetheless, a renewed and wider interest in NM was brought about in the last decade by the successes in AI and DL. Recent research and development efforts on NM sensors, processors and algorithms have demonstrated the very promising potentials of NM systems¹² in many fields of application. It is clear that the Security and Defence domain could definitely benefit from NM systems, as many of its applications present significant SWaP and timing constraints. In particular, the NM chip market is predicted to reach \$1bn by the year 2025,¹³ with Aerospace and Defence being amongst the leading industries for this growth. For example, the US government has been heavily investing in NM technologies for more than a decade now, through DARPA and the SyNAPSE programme.^{14,15}

This paper reports on the current state of the art in the field of NM systems, in terms of algorithms, sensors and processors, and it describes three application scenarios of SNN-based processing for security and defence, namely target detection and tracking, semantic segmentation, and control. In the first example reported, SNN are used to detect and track objects through using a Convolutional Neural Network architecture, to extract hierarchical feature representations at differing spatial regions. This results in a successful detection methodology, while using the latent space spatial information to give a rough estimate of location. As second example, previous work has shown that SNN-based processing can be used to segment spike-based images. This builds upon the detection/tracking methodology described, by constructing an encoding-decoding network. This allows the same level of detection capability now with a precise pixel spatial location. A third application of SNN is in the field of control. This can exploit a spiking event-based processing pipeline to give low latency control outputs. When coupled with the segmentation network, the controller is then able to give control updates based off the spiking location data out from this network. As demonstrated by the case studies described in this paper, the field of NM engineering has great potential to revolutionise data acquisition and processing in many fields of application. Moreover, NM is perfectly suited to defence and security applications, where the SWaP profile is a fundamental driving factor in many design and development tasks.

The remainder of this paper is organised as follows. Section 2 introduces NM processing in its three components, i.e. algorithms, sensors and processors. Section 3 describes three case studies that showcase the potentials and benefits of NM systems. Section 4 finally concludes the paper.



Figure 1. Modelling of the synaptic connection between two spiking neurons, with the information representation by the train of spikes traveling along the synapse, from the pre-synaptic neuron to the post-synaptic neuron.

2. NEUROMORPHIC PROCESSING

Neuromorphic systems aim to closely mimic the human nervous system, in software and/or hardware and circuitry. This represents a new paradigm both in terms of sensing and processing. Therefore a new typology of sensors is required, to directly encode the information into spikes. At the same time, novel algorithms are necessary, to fully and purposely exploit the new nature of the data. Finally, new processing architectures are fundamental, to optimise execution.

2.1 Algorithms

ANNs were initially inspired by the human nervous system, even though some simplifications had to be made. In this context, as the human brain is asynchronous, very efficient, consumes very low power and works with spikes, researchers have proposed the use of SNNs to mimic such key characteristics, with the aim of solving some of the aforementioned issues associated with DL architectures.

SNNs are a particular type of neural networks where the information is indeed encoded in the timing and rate of spikes.¹⁰ Neurons in a typical ANN would use floating point values to ensure high precision, while an SNN only uses 1 bit, i.e. binary, data: spike or no spike. The information is encoded into spikes,^{16,17} within emission time (when) and frequency (how often) of the spikes,¹⁸ and not in the amplitude, as this is constant. In essence, only a small amount of important information passes through the network, but in a timely manner; moreover, such information is all fully relevant, with no redundancy.

SNNs are therefore inspired by information processing in biology, where sparse and asynchronous binary signals are communicated and processed in a massively parallel fashion. Alongside low power consumption, SNNs also provide fast inference and event driven information processing. SNNs can be seen as the algorithmic part of NM.⁹

2.1.1 Spiking neuron model

A spiking neuron is very similar to the perceptron of an ANN. The main difference is represented by the fact that a spiking neuron generates a spike in output if the neuron membrane potential exceeds a certain threshold, while the output of the ANN perceptron is determined by the particular activation function chosen.

The biologically inspired synaptic communication between consecutive neurons is modelled as a link that carries the train of spikes from a pre-synaptic neuron to a post-synaptic neuron (Figure 1); this link has an associated synaptic weight W , which is combined with the post-synaptic neuron membrane potential, referred to as post-synaptic potential (PSP). Each input spike causes a charge in the neuron's membrane potential, and if the threshold is exceeded, an output spike is fired by the post-synaptic neuron.¹⁹

There are various spiking neuron models proposed in literature, with the most commonly adopted being Hodgkin Huxley²⁰ and Leaky Integrate and Fire (LIF).²¹ The difference in such models is how the PSP evolves before and after the generation of the output spike.

In the LIF neuron, the PSP follows the behaviour illustrated in Figure 2a. As each pre-synaptic spike increases the neuron's potential by its synaptic weight, the neuron's potential is the summation of all weighted input PSPs

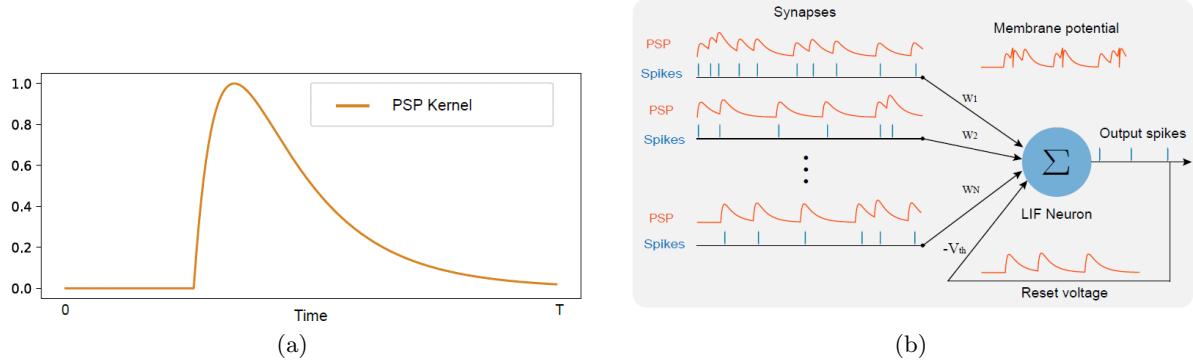


Figure 2. (a) PSP behaviour in a LIF spiking neuron; (b) illustration of how the spiking neuron membrane potential is affected by the spikes present on all the synapses of the neuron.¹⁹

plus a negative voltage, which serves as a reset mechanism at the time of spike. A neuron accumulates all input PSPs, such that the membrane potential is as illustrated in Figure 2b.

2.1.2 Learning through Spike Time Dependent Plasticity

Although backpropagation is in general the de facto standard algorithm to train ANNs and DNNs, directly adapting backpropagation to an SNN scenario is not feasible because the output of a spiking neuron is a Dirac delta function at spike time and 0 at all other times, hence it is non differentiable. One approach to address this problem is to train an ANN and convert the model to an SNN. Despite being a straightforward method, such a conversion requires additional fine tuning of weights and thresholds. Moreover, it does not really allow for a ground-up design of the final SNN, and therefore the core advantages of the spiking paradigm might not be fully exploitable. Researchers have proposed alternative approaches to emulate the backpropagation algorithm in an SNN.^{22–25} However, in this case the drawback is in the higher complexity of such a network, which again might lose some of the SNN advantages.

A more suitable approach to train SNNs is represented by the biologically-inspired Spike Time Dependent Plasticity (STDP) mechanism.^{21, 26} In biology it has been seen that the timing between a pre-synaptic spike and a post-synaptic spike can induce persistent synaptic modifications.²⁷ As illustrated in Figure 3a, spikes travel through synapses and arrive to the post-synaptic neuron in an asynchronous fashion. As this happens, the PSP in the post-synaptic neuron changes, i.e. it is increased for each spike by the weight associated with each synapse (W_1 , W_2 and W_3). When the PSP passes the neuron threshold, an output spike is generated, and it is assumed that such a spike has a semantic relationship with the input spike that caused the neuron to fire (Figure 3b). This is encoded by an increase in the synaptic weight W_2 , to signify that there is a cause-effect relationship between the latest input spike and the output spike. At the same time, the other synaptic weights (W_1 and W_3) are either increased or decreased, depending on their own time interval from the triggering spike (Figure 3c).

In simple terms, if a pre-synaptic spike is quickly followed by a post-synaptic spike, it is assumed that these two events, i.e. the spikes, are related, and therefore the synapse is modified to represent this. The synaptic strength, i.e. the synaptic weight, is increased if a pre neuron subsequently causes the connected post neuron to fire and the difference in the respective spiking instants is small, which signifies a strong causal relationship between the corresponding pair of neurons. On the other hand, the synaptic strength is decreased for larger spike time differences. In mathematical terms, this can be expressed²¹ as in (1):

$$\Delta w_{STDP} = \eta \times \left[\exp\left(\frac{t_{post} - t_{pre}}{\tau}\right) - offset \right] \times [w_{max} - w]^{\mu} \quad (1)$$

where Δw_{STDP} is the increase or decrease in the synaptic weight, η is the learning rate, t_{pre} and t_{post} are respectively the times of the triggering spike and of the output spike, w_{max} is the maximum synaptic weight (equal to 1), w is the previous weight value, and μ is an exponential factor. An example of the increase of the

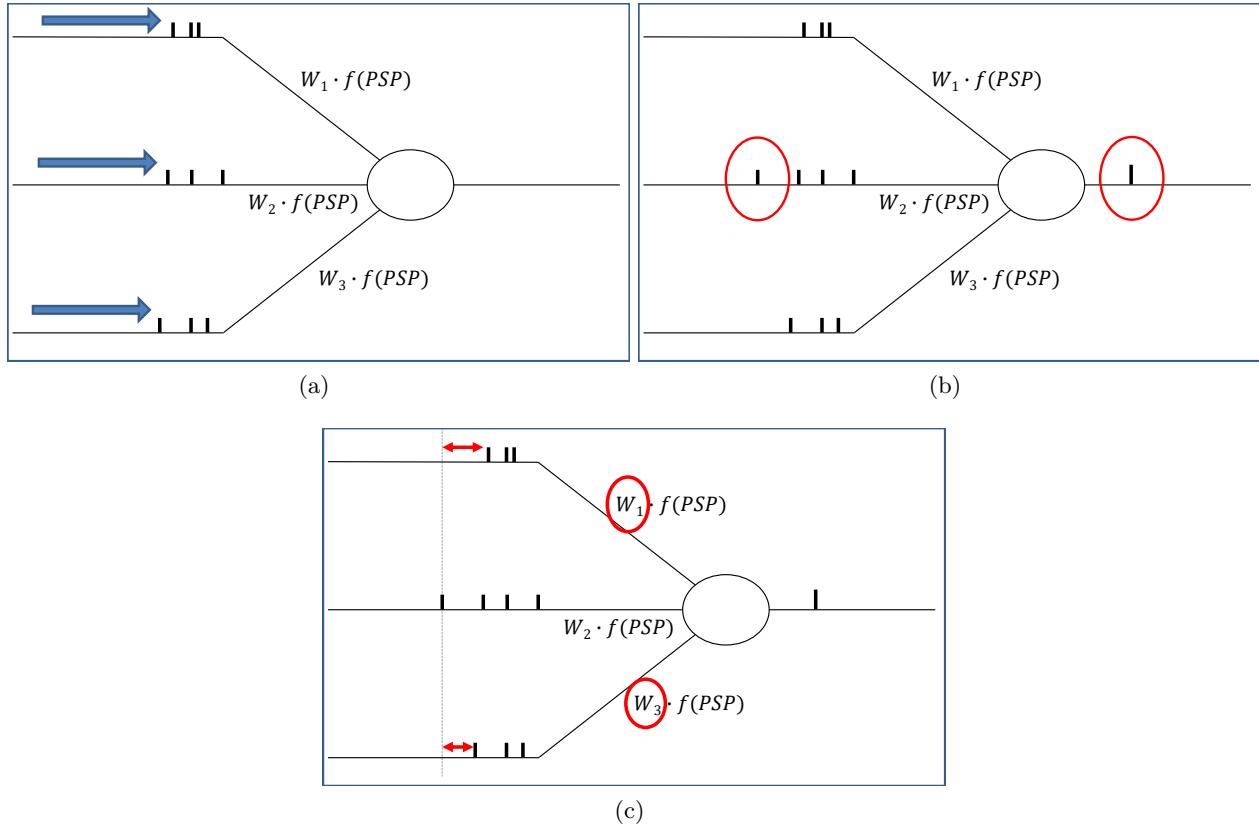


Figure 3. Visual representation of the timing aspect of the STDP mechanism.

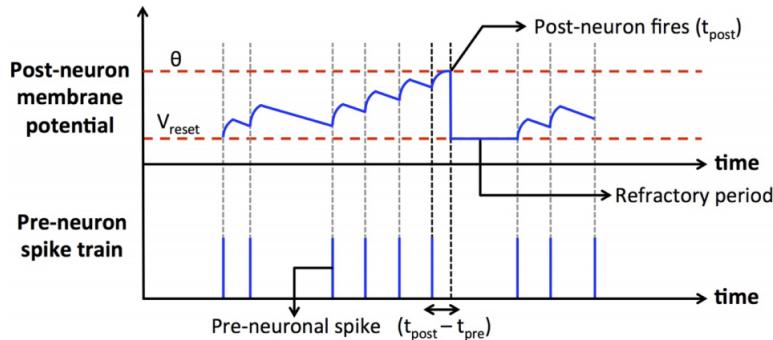


Figure 4. The membrane potential is increased at every new spike occurring, until a threshold is passed and so an output spike is generated. Then the membrane potential is decreased to a reset value, to avoid constant firing.²¹

PSP is illustrated in Figure 4, which also includes the refractory period wherein the post-synaptic neuron is inhibited from firing.

As in an ANN, the synaptic weights encode the ‘learning status’ of the network. However, an SNN that uses STDP is a feedforward network, as there is no need for backpropagation. Spikes travel through the synapses asynchronously, i.e. as they happen, and so time and timing are therefore two important factors in the spiking behaviour of a neuron. STDP is an unsupervised learning method, as the synaptic weights are updated ‘on the fly’, as the data passes through the network, without a direct link to a ‘ground truth’. The information encapsulated in the relative timing of the spikes is automatically extracted by the network and encoded in the values of the weights. An SNN simply ‘learns’ important features, and ideally the network is always learning, unless the learning rate is set to zero.

2.2 Sensors

To maximise the benefits provided by SNN architectures, the data in input must be in a suitable spiking format. While it is possible to convert conventional data into spike-based representations, the acquisition rate of conventional non-spiking sensors would remain the limiting factor in terms of speed and throughput in a NM system. Therefore real spiking sensors are required to achieve high rate, low latency, and high dynamic range. For example, event-based vision sensors are biologically-inspired NM cameras, which mimics how the human eye perceives motion within the retina.²⁸ This replaces the fixed frame rate of traditional camera images, with a far more compressed and sparse output, resulting in 1 to 3 orders of magnitude increase in output rate. This allows the sensor to have a much higher temporal resolution, but without the caveat of the extra processing required for the pixels that did not change.

In terms of data, conventional vision, audio and electromagnetic sensors generate large volumes of redundant data sampled at a constant rate that, when processed through such complex DL structures, tend to consume significant levels of power and storage. For example, for object detection in a relatively cluttered scenario, there may be many objects to detect, so it makes sense for a CNN to process the entire image. On the other side, in a very sparse scenario, as for example detection and tracking of a drone in the sky, the CNN still has to go through each pixel in the image, even if quite clearly it is very sparse and there is only one object to detect. In such a scenario, an event-based camera would only highlight the changes in the scene caused the drone motion, while the vast majority of the image does not change and therefore no further information is generated.

NM sensors change the way in which the world is sampled by producing spikes as output, in an asynchronous and event driven manner. The event-based nature of NM sensors can be viewed as a paradigm shift in how sensing is achieved: emphasis is on precise timing, while being scene and context dependent, such that the amount of activity being sensed relates to the amount of information output.

A typical example of NM sensor is represented by the Dynamic Vision Sensor (DVS) developed by IniLab.^{29–31} The DVS is a biologically-inspired sensor widely used in the research community, and it asynchronously transmits the logarithmic light intensity difference, as events, on a pixel by pixel level. The output from the DVS consists of spikes with the associated time and position, and they are not constrained to a fixed frame rate, as in traditional cameras. Therefore the output of the DVS camera is far more compressed and sparse. In this way the sensor has a much higher temporal resolution, equivalent to a 66000 frames per second super slow-motion camera for up to 800 pixels. Another useful feature is the DVS's high dynamic range, rated at >120dB vs the <60dB of traditional cameras.³² This allows this event-based camera to see in a wide variety of lighting conditions, from quickly changing brightness conditions, to low light ones, where traditional cameras would not be able to detect anything. Examples of visual output from the DVS camera are illustrated in Section 3.

Other event-based cameras currently available on the market are from Prophesee³³ and from Samsung.³⁴ Researchers are also working on other types of spiking sensors, as for example a NM olfactory circuit,³⁵ and a NM skin.³⁶

2.3 Processors and software tools

NM engineering attempts to reproduce some of the unparalleled computational power and efficiency of the brain, by mimicking its function and structure. This results in a number of hardware and software implementations which aim to act as a massively parallel accelerator for SNNs. Since the neuron spikes within such a processor are inherently asynchronous, a NM processor is the best computational partner for the sensors described in the previous Section 2.2. Combining this type of sensors and processors directly offers greater gains to be realised in locality, both temporal and spatial, which in turn can help the computing systems to yield lower power and latency.

Neuromorphic processors are split into two main categories, that being analog and digital, with the main processors of analog being Braindrop³⁷ (Neurogrid³⁸), BrainScaleS³⁹ ROLLS⁴⁰ DYNAP-se⁴¹ while the digital ones are TrueNorth,⁴² Loihi,⁴³ ODIN⁴⁴ and SpiNNaker⁴⁵ (although SpiNNaker is a software based implementation). Another notable factor in differentiating the processors is the capacity for on-chip learning which is exclusive to Loihi, ODIN and DYNAP-le.⁴¹

Together with the hardware, there is also an importance for usability in terms of software tools, compilers and APIs. SpiNNaker aims to simulate a biologically realistic models of the human brain and uses general-purpose ARM cores to do so.⁴⁵ This software implementation of the neurons allows the most amount of flexibility, permitting any type of neuron model, but at the cost of processing speed. TrueNorth makes use of digital neurons to perform real-time inference. Each chip can simulate up to 1 million neurons and 256 million synapses, distributed among 4096 neurosynaptic cores.⁴² The networks for TrueNorth are trained offline using a GPU or other processor; then the weights are ported onto the chip itself. Loihi can perform real-time inference and online learning from the use of digital neurons. Each Loihi chip can simulate up to 131 thousand spiking neurons and 130 million synapses. Multiple chips can also be stacked together in a system called Pohoiki Springs with 768 Loihi chips, resulting in 100 million neurons and 100 billion synapses. Rules such as STDP and reinforcement learning can be implemented due to the learning engine within each NM core allowing updates of each synapse.⁴³ Non-spiking variants of networks can be trained using TensorFlow and converted to a spiking networks for Loihi using the Nengo⁴⁶ Deep Learning toolkit from Applied Brain Research.⁴⁷ The Dynamic Neuromorphic Asynchronous Processor (DYNAP) from SynSense comes in two variants, one designed for scalable inference (Dynap-se), and the other for online learning (Dynap-le).⁴¹ Braindrop prototypes a single core, the Brainstorm system,³⁷ of the 1 million neuron; it is programmed using Nengo and implements the Neural Engineering Framework.⁴⁸ Braindrop is also the successor to the Neurogrid processor.

3. CASE STUDIES

The three case studies reported in this section provide an overview of some end-to-end NM approaches, i.e. from spiking input to SNN-based output. This methodology enables the greatest benefits from the characteristic of NM sensors to be carried through to the SNN processing. Such a methodology ensures asynchronous data flow, which maintains temporal features present in the data while also conserving temporal sparsity. It is through exploitation of these characteristics that benefits such as less computational power required, more efficient and faster processing, and much lower power consumption, can come to fruition. The following case studies aim to highlight some of the key benefits of this NM approach within the defined problem area.

3.1 Detection

Simple methods for detection and classification were shown in Kheradpisheh et al.,⁴⁹ and then utilised on UAVs within the following paper by Kirkand et al.⁵⁰ The latter proposes a low latency method of detection, that could successfully detect UAVs in a number of scenarios including adverse lighting conditions. The paper also highlights that the method can work with data direct from a NM vision sensor (in this case the DVS 240) and with video converted into a spiking format, using a difference of Gaussian's filter to convert the frames of the video into a temporal spike pattern based on intensity. This paper highlights two key benefits of a NM approach, one from the sensor, that is high dynamic range, and the other in the processing, in the form of asynchronous operation.

The dynamic range is independent per pixel, and this allows the sensor to have a higher overall dynamic range of around 130dB vs the 60dB of a traditional camera.³⁰ This characteristic of the sensor can be best seen when dealing with scenes that have bright light sources and large dark regions, in which a traditional camera would chose a dynamic range within the bright of dark regions and therefore over- or under-exposing the rest of the image. An example of this high dynamic range benefits is shown in Figure 5, where a well lit scene is shown beside a poorly lit one, to demonstrate how difficult it is to notice the UAV in this second scene (the UAV is circled in the image to highlight it). This is in contrast to the NM sensor, which manages to deliver very similar results once filtered. This highlights the sensors ability to work within a wide range of lighting conditions, without having to make any adjustments to the sensor. This gives the sensor an attribute that is typically reserved for specialist sensors or from multiple exposure blending techniques. The asynchronous operation provide dynamic relationship to the visual source. In other words the output of the sensor is directly proportional to the amount of movement within the scene. This attribute determines how the sensor can deliver a sparse yet detailed account of the scene, minimising computation and power. An example of this is shown in Figure 6, where a UAV flying within two scenes is pictured. The two scenes depict a fast moving and slow moving target, and the accumulation of the sensor output over a range of different time intervals (10-200ms)

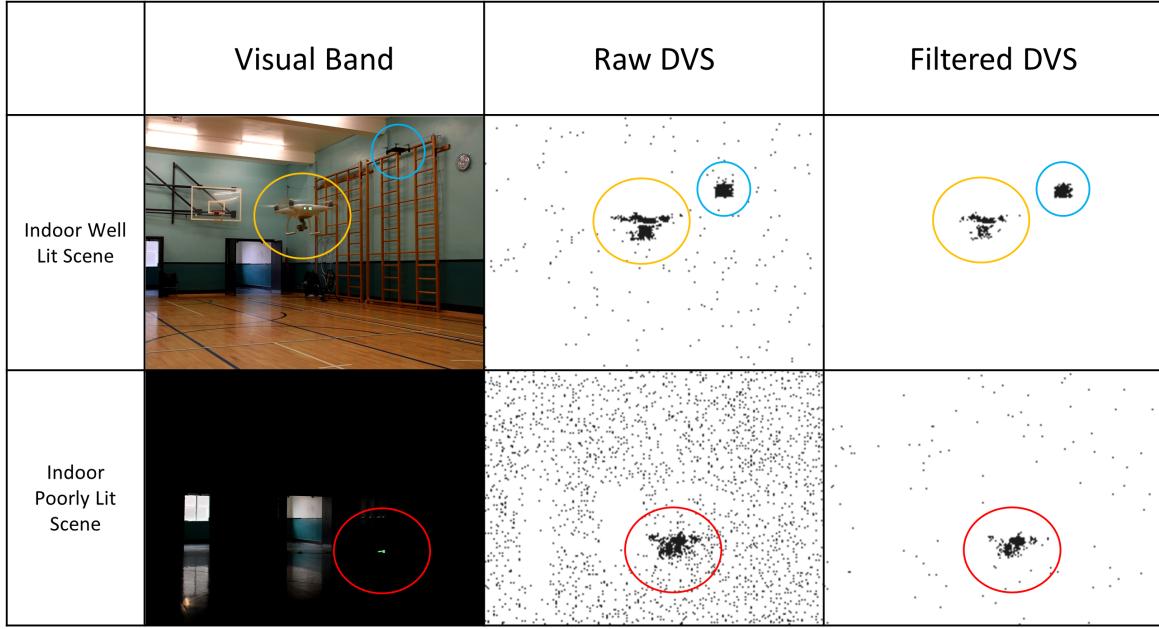


Figure 5. Illustration of the benefits of the sensor in low light conditions.

is shown. This highlights the ability of a NM sensor to capture highly dynamic scenes with detail, while also displaying the sparsity brought about through the removal of redundant repetitive information within a slow moving or stationary scene. This ability to produce a high volume output for dynamic scene and sparse output for static scene is a key factor in the reduction of computational overhead. It is these attributes that the NM processing stage aims to maintain throughout.

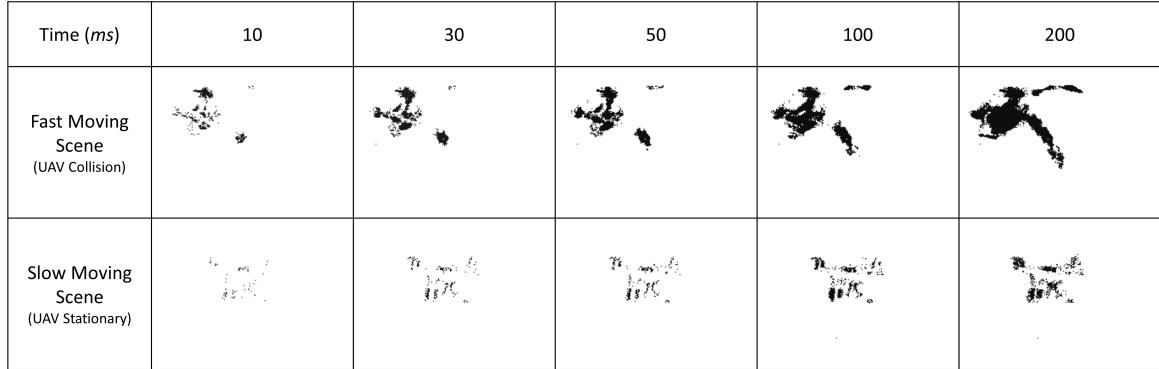


Figure 6. Illustration of the benefits of the sensors in dynamic environments.

The processing stage utilises a SCNN in order to maximise the benefits gained from the NM sensor, where the detection method utilises a layer-wise unsupervised learning approach STDP. This allows the network to pick out reoccurring patterns in hierarchical format utilising the convolutional nature of the network, to build more complex patterns through the networks layers. An example of this feature extraction process is shown in Figure 7, where Layer 1 starts off with simple Gabor like edge features, then builds into more complex shapes in Layer 2, before being used for classification in Layer 3. To help further visualise the features being used within the network, an illustration of the features overlaid onto the original input is shown in Figure 8, which shows

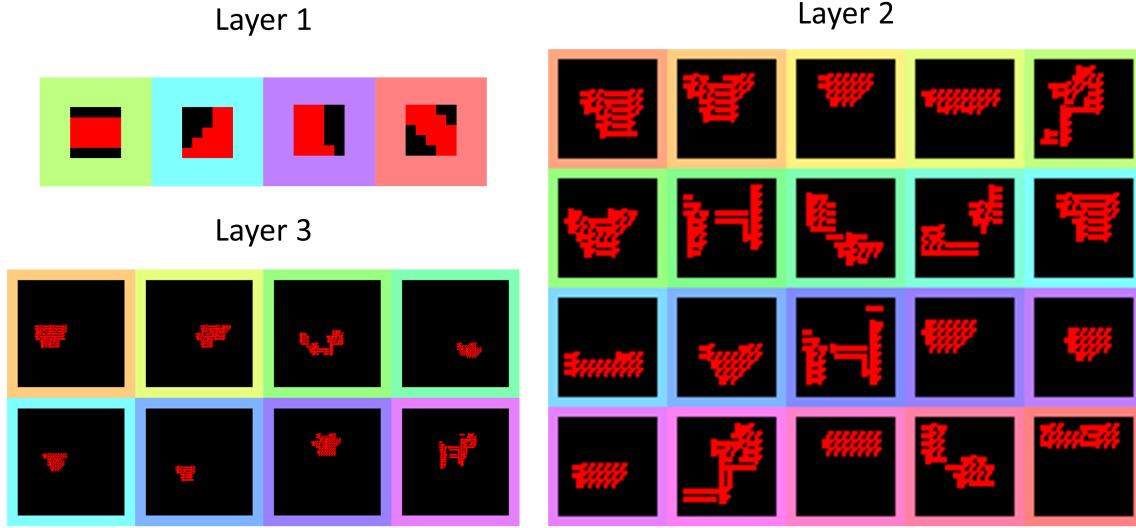


Figure 7. Illustration of the learn features of each of the three layers of the network for UAV detection.

the colour coordinated features and how they relate to the original input. Within this network the CNN brings



Figure 8. Illustration of the features learned from the network mapped onto the original input showing the location of origin and how the features are used to make the classification.

a local spatial coherence and parameter/weight sharing method, as typically seen in DL implementations, that allows an image to be compressed into a latent space, such that it can be represented by a respectively smaller number of features compared to the number of pixels in the image. At the same time the SNN allows sparsity to continue through to the processing with a spiking neuron model, and sparsifying winner takes all learning mechanism.

Overall the network was able to deliver 91% accuracy for detection within a wide range of scenes, all while

delivering over 200 times deduction in computational overhead thanks to the combination of the sparse sensor and processing. This is highlighted with the fact that only 200 pixels were required to successful detect a UAV in the scene, which is 0.5% of the overall sensor needing to be active. This sparsity can deliver many benefits with reductions in computational processing, leading to a reduction in overall size, weight, power and cost, and therefore improving overall application system viability.

3.2 Semantic segmentation

Semantic segmentation of spiking based images through the use of a spiking fully convolutional encoding decoding network⁵¹ can be seen as the next progressive step of a detection based system.^{49,50} This novel use of an encoder-decoder structure with learned weights from an unsupervised STDP approach allows low latency asynchronous output of semantically segmented images. Within a semantic segmentation network, the learned features of the STDP approach similar to that seen in the previous case study are now used in post classification. These features are utilised through the decoding network to map the latent feature space back onto the original pixel space. This allows a precise sparse segmentation to occur, with sparse being in the sense that the segmentation is only made up of the most salient features used throughout the encoding process to help classify. An illustration of this type of network and the input to segmented output is shown within Figure 9. The figure highlights how the input after classification and decoding is then turned into a sparse segmented version of the input, highlighting the features that most resembled that class. Throughout testing of this network, a high accuracy for classification was paired

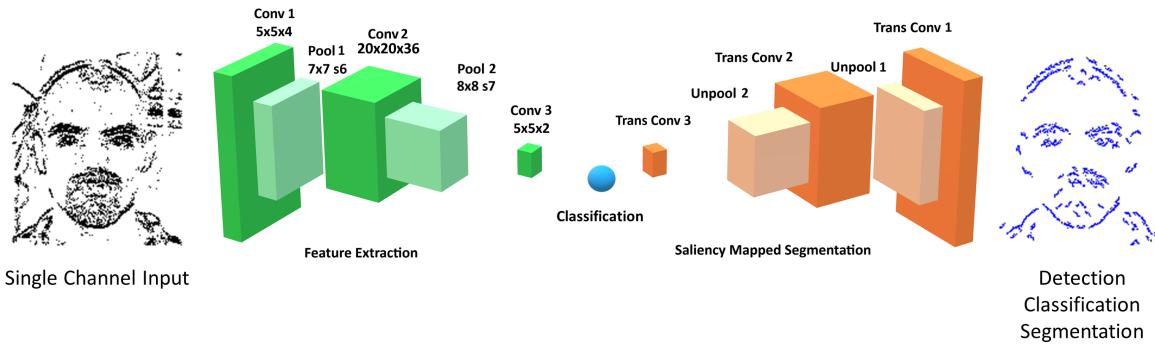


Figure 9. Illustration of the network used within Kirkland et al.⁵¹ with the input and segmentation output shown to the left and right of the network.

with a high overall mean intersection over union, which is a measure of how well the segmentation process was carried out. The idea behind this network was to provide a low latency semantic segmentation process to the NM domain. Classically segmentation tasks are computationally expensive and involve some sort of regression task to align the possible multiple instances of classification. Within this method, an exploit of the saliency mapping potential of SNN was utilised to create temporal spiking activity maps, that permit the decoding layers to perform a transposed convolution process, to map the classification back into pixel space. Within this work converted and actual NM sensor data was utilised on the same dataset to give perspective of the difference in fidelity of the convert images compared to the actual NM sensor (a DVS 240 with 240x180 pixels) which used the N-Caltech dataset,⁵² a recorded version of the Caltech dataset⁵³ off of a screen.

The segmentation of the network was tested across three scenarios, the two previously described with a Caltech converted with a difference of Gaussian's filter and the N-Caltech dataset. The third scenario involved two object that partially occluded each other in order to test the occlusion threshold of the proposed approach. Throughout the testing, the network maintained above 90% accuracy, while scoring 74%, 68% and 81% on average on the mIoU metric, across all the tested input streams. To visualise how the network is taking the

encoding structure shown in Figure 9 and mapping it into pixel space, a visualisation of the processing stage is illustrated in Figure 10. Figure 10 highlights the process undertaken using the learned weights to convert

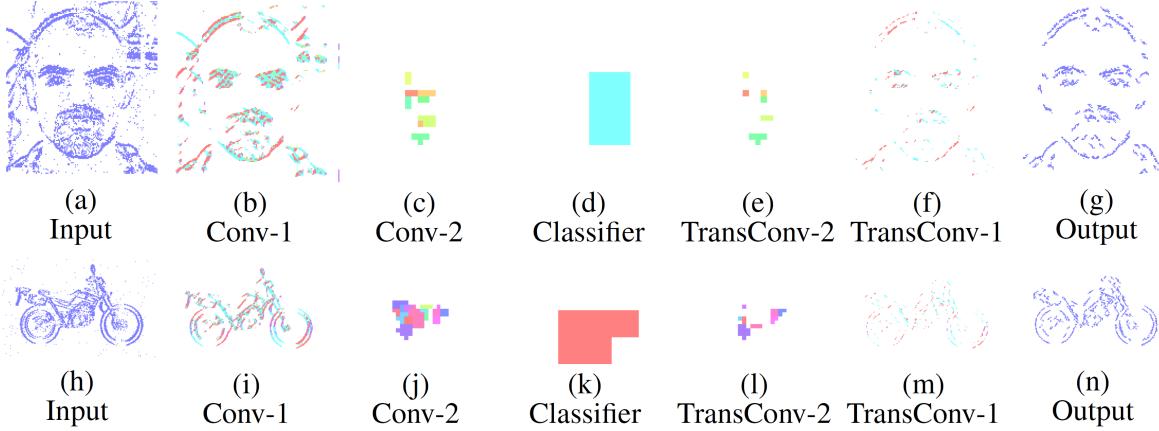


Figure 10. Illustration of the processing stage of the network taking the input and through the learned features mapping it into a latent space for classification, then through the use of the temporal spiking activity mapping is able to decode it back to pixel space.

the spiking input into a stream of features over time, and then back into the pixel space through the decoding process. The image itself is an accumulation of the features and active pixel over time to help with visualisation. It can also be seen in Figure 10 that the features represented in (c) and (e) have different colours from that of (j) and (l). This, similarly to the previous case study, is linked to the feature representation, which is shown in Figure 11, illustrating the learned features of the network. It is also apparent that Figure 11 learns two different sets of features for the face and motorbike classes prior to the third layer. It is also very visible that the feature extraction process through this method of learning with STDP is much more intuitive and interpretable than CNN training through backpropagation. Similar results were also found for the NM sensor data, although with

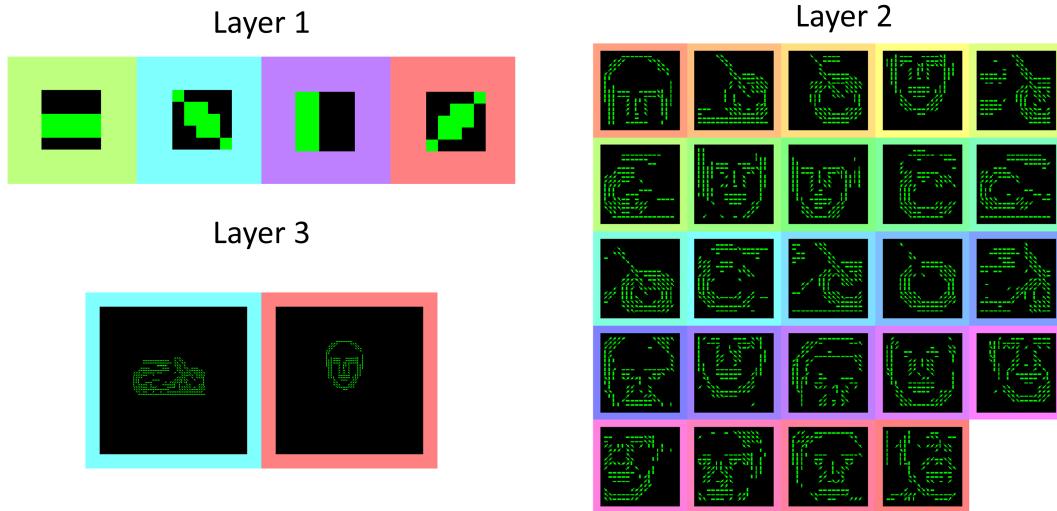


Figure 11. Illustration of the learn features of each of the three layers of the network for face and motorbike detection and segmentation.

a slight dip in both the accuracy and mIoU values, due to the imperfect process of recording the dataset off

a screen with micro-movements of the camera. To allow better understanding and testing of the NM sensor, a small dataset capture with two desk ornaments allows testing of segmentation and segmentation in face of partial occlusion. The network is able to successfully decide what features belonged to each of the classification made, thanks to the temporal spiking activity maps which allow the untangling of the semantic classes through correctly mapping the salient features as shown in Figure 12, with a large input to output image shown in Figure 13. The images in Figure 12 show the encoding process at the top, with the lower three lines representing the two separate object classes, individual segmentation and decoding process, while the last line illustrates the combined segmentation and decoding process. Also, Figure 13 highlights how the combined features look when coloured according to class and compared with the input stream. Overall this work was able to successfully

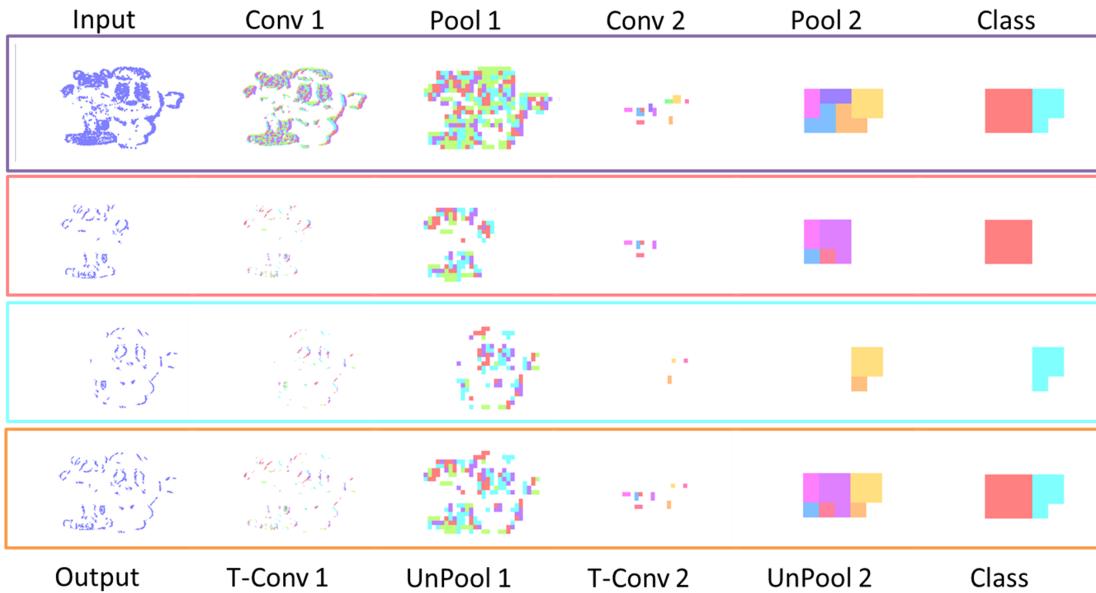


Figure 12. Illustration of the semantic segmentation with occluded objects, highlighting the features belonging to each class in the process with the top line showing the encoding process with the lower three lines showing each class and a combined segmented image and its associated decoding process.

show that semantic segmentation could be performed in a NM manner, allowing the exploitation of asynchronous processing, low latency and less computation throughput, when used within a NM system.

3.3 Tracking and Control

The last case study highlights the results from the use of the previous two, together with an action based tracking and control system,⁵⁴ where the previous segmentation methodology is now built into an end-to-end NM system for following a chosen class within a scene. This describes how to use the results of the semantic segmentation to help to provide a control output for a system, such a camera on a pan tilt mount, that could be used to tracking a class as it moves around a scene.

A fully NM system should be able to benefit from the sparse asynchronous characteristics throughout the whole pipeline. This would allow the greatest benefits to be achieved in terms of less computational power required, more efficient and faster processing and much lower power consumption. An overview of such a system is shown in Figure 14 which depicts a Perception, Understanding and Action system, where each elements is essentially a stand alone NM solution; however, when combined, they achieve the greatest benefits. The perception and understanding parts of the system come from the previously mentioned case studies, but additional features have been added to help improvement performance in a range of situations. The new system has a new pre-empt and adapt thresholding approach, allowing the understanding section of the system to deal

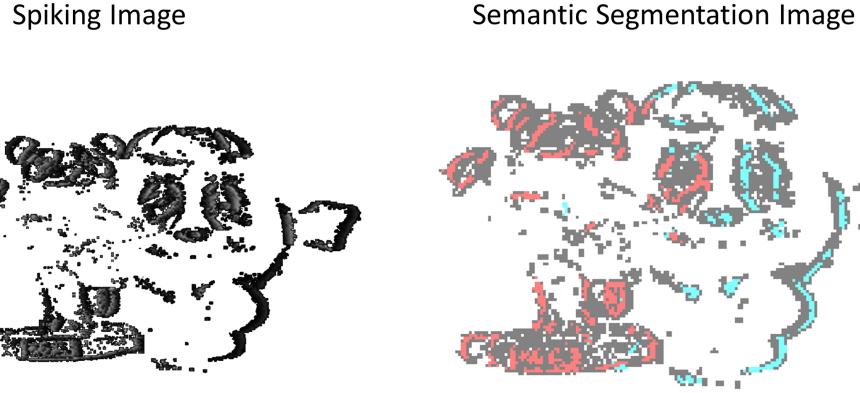


Figure 13. Illustration of the input and associated semantically segmented image to show the detail of the segmentation and class association even within the occlusion areas.

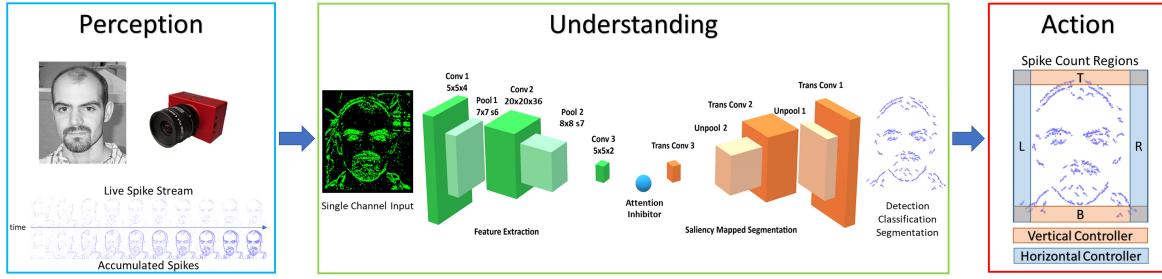


Figure 14. Illustration of proposed framework for perception understanding action.

better with input noise. This new implementation is highlighted in Figure 15, which highlights the difference between a normal (a) and noisy (b) input image and the segmentation. This is then dealt with first by the pre-emptive (c) approach raising the threshold to not allow erroneous segmentation, before the adaptation (d) allows successful segmentation. This new thresholding method is also joined with a feature pruning method that allows underutilised features to be reset during the training process, in order to learn a better and more representative feature set. This is depicted within Figure 16, highlighting the resetting of features that had a low rate of convergence and under or over active spiking compared to other features, where the original features are shown in (a), then reset in (b) before learning new features in (c). Together these new features allowed the understanding system to improve upon previous iterations. The network was now able to achieve better results, i.e. 96% accuracy and 81% IoU score on the previously shown dataset. Further to this, the network was able to be extended to segmentation successfully to more classes, showing a scalability within the network.

From these successful improvement, a set of experiments were ran to envisage how an action (control and tracking) system could be added to the segmentation. This was realised through a monitoring of the periphery of the input stream post segmentation. Any high count of spiking activity within these regions would directly output a control signal to a pan-tilt system, that would allow tracking of successfully segmented classes, as it moved within the scene attempting to move out the field of view. The results of the experiment are shown in Figure 17, where (a) shows the input sequence, (b) shows the classification being center of the field of view so no control signal, (c) shows the segmentation entering the rightmost region causing a control output to pan right, while (d) shows the segmentation on the leftmost of the field of view causing a control signal to pan left.

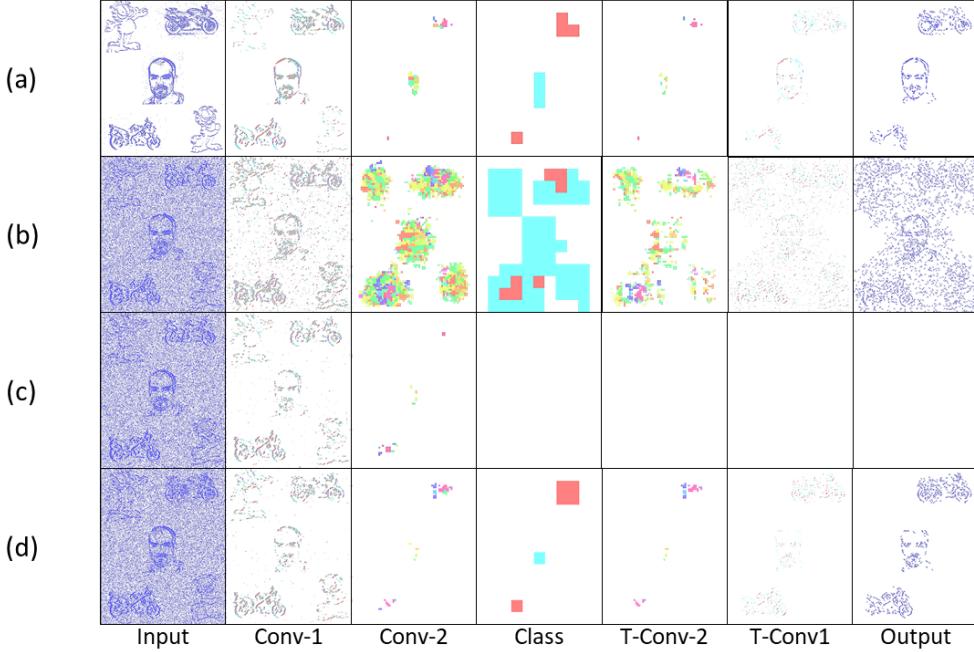


Figure 15. Illustration of thresholding dealing with excess noise. (a) Normal image processing and segmentation, (b) Noisy image processing and segmentation, (c) Pre-Emptive thresholding increase and (d) Adaptation allowing adequate thresholding for successful segmentation.

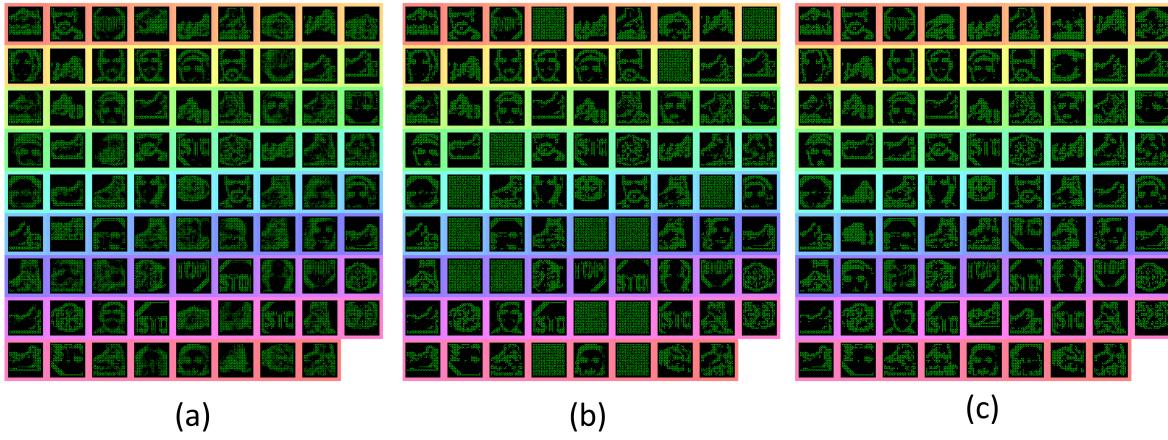


Figure 16. Illustration of pruning with features being reset to allow better feature representation.

Overall the case studies present recent improvements within the realm of computer vision based tasks. These case studies provide an overview from a Hebbian learning based approach utilising only local learning rules, where backpropagation in its standard form would be prohibited. However, there are other lines of research that utilise this deep learning type approach, followed by a conversion to spiking neurons. The issue with this approach is that some of the novel features of SNNs are lost in this manner, including online-learning or adaptations along with the loss of some of the interpretability. This then presents a spectrum of solutions which can be applied in a problem dependent solution, where the required feature set will help to guide the type of NM approach required.

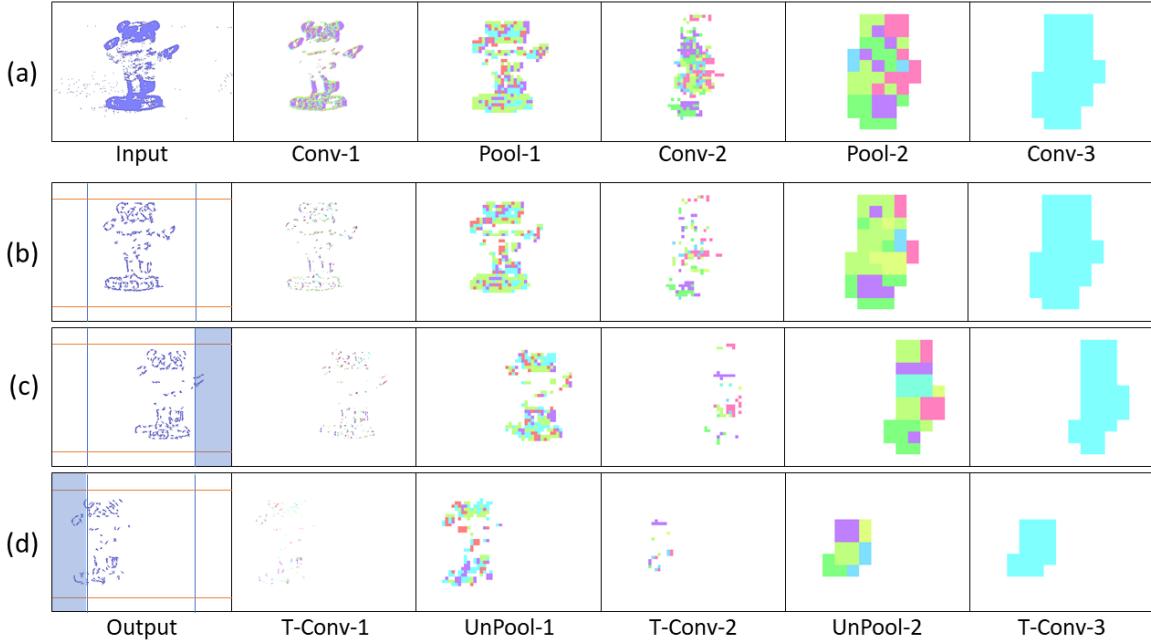


Figure 17. Illustration of tracking through control signal from periphery of image.

4. CONCLUSION

As reported above, there are great expectations on AI and NM in many fields of application. Although in some cases such expectations somehow exceed what is currently possible from a technological point of view, this should not be seen as a negative outlook, as in recent years DL has impressed a considerable acceleration to the effective application of ML to real life problems and scenarios, in ways that would have been difficult to foresee 10-15 years ago. In this context, NM systems are also rapidly gaining momentum, with many industries and governments behind them, in the form of considerable investments of funding, time and effort in these technologies. In particular, there are clear opportunities for NM in the security and defence domain, in all those applications where there are SWaP and timing constraints, or the sensed data has an inherent spiking format. And more applications are likely to arise, as technologies evolve.

ACKNOWLEDGMENTS

The authors would like to thank Leonardo for funding the research work that led to the three case studies presented in this paper.

REFERENCES

- [1] Duda, R. O., Hart, P., and Stork, D. G., *[Pattern Classification]*, Wiley, 2nd ed. (2000).
- [2] Goodfellow, I., Bengio, Y., and Courville, A., *[Deep Learning]*, MIT Press (2016). <http://www.deeplearningbook.org>.
- [3] Schmidhuber, J., “Deep learning in neural networks: An overview,” *Neural Networks* **61**, 85–117 (2015).
- [4] Patterson, J. and Gibson, A., *[Deep Learning]*, O’Reilly Media, 2nd ed. (2017). <https://www.oreilly.com/library/view/deep-learning/9781491924570/>.
- [5] “Spring budget 2017.” https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/597467/spring_budget_2017_web.pdf. Accessed: 02/06/2020.
- [6] “UK’s research and innovation bodies welcome budget.” <https://www.gov.uk/government/news/uks-research-and-innovation-bodies-welcome-budget>. Accessed: 05/06/2020.

- [7] “Budget 2017: Robotics, driverless cars and 5G to get UK funds.” <http://www.bbc.co.uk/news/technology-39203733>. Accessed: 05/06/2020.
- [8] “Artificial Intelligence – Shaping Europe’s digital future.” <https://ec.europa.eu/digital-single-market/en/artificial-intelligence>. Accessed: 19/08/2020.
- [9] Roy, K., Jaiswal, A., and Panda, P., “Towards spike-based machine intelligence with neuromorphic computing,” *Nature* **575**, 607–617 (2019).
- [10] Tavanaei, A., Ghodrati, M., Kheradpisheh, S. R., Masquelier, T., and Maida, A., “Deep learning in spiking neural networks,” *Neural Networks* **111**, 47–63 (2019).
- [11] Mead, C., “How we created neuromorphic engineering,” *Nature Electronics* **3**(7), 434–435 (2020).
- [12] Pfeiffer, M. and Pfeil, T., “Deep Learning With Spiking Neurons: Opportunities and Challenges,” *Frontiers in Neuroscience* **12**, 774 (2018).
- [13] <https://www.globenewswire.com/news-release/2020/04/25/2022050/0/en/Global-Neuromorphic-Chip-Industry.html>. Accessed: 11/08/2020.
- [14] <https://www.nextplatform.com/2017/06/26/u-s-military-sees-future-neuromorphic-computing/>. Accessed: 11/08/2020.
- [15] <https://www.darpa.mil/program/systems-of-neuromorphic-adaptive-plastic-scalable-electronics>. Accessed: 11/08/2020.
- [16] Gerstner, W. and Kistler, W. M., “Spiking Neuron Models: Single Neurons, Populations, Plasticity,” (2002).
- [17] Gutig, R. and Sompolinsky, H., “The tempotron: a neuron that learns spike timing-based decisions,” *Nature neuroscience* **9**(3), 420–428 (2006).
- [18] Krause, T. U., *Rate Coding and Temporal Coding in a Neural Network*, Master’s thesis, Ruhr-University Bochum (2014).
- [19] Fang, H., Shrestha, A., and Qiu, Q., “Multivariate time series classification using spiking neural networks,” in [*International Joint Conference on Neural Networks IJCNN*], (2020).
- [20] Hodgkin, A. L. and Huxley, A. F., “A quantitative description of membrane current and its application to conduction and excitation in nerve,” *The Journal of physiology* **117**(4), 500–544 (1952).
- [21] Panda, P., Srinivasan, G., and Roy, K., “Convolutional Spike Timing Dependent Plasticity based Feature Learning in Spiking Neural Networks,” (03 2017).
- [22] Shrestha, S. B. and Orchard, G., “SLAYER: Spike Layer Error Reassignment in Time,” in [*32nd Conference on Neural Information Processing Systems (NeurIPS 2018)*], (2018).
- [23] Shrestha, A., Fang, H., Wu, Q., and Qiu, Q., “Approximating Back-Propagation for a Biologically Plausible Local Learning Rule in Spiking Neural Networks,” in [*Proceedings of the International Conference on Neuromorphic Systems*], Association for Computing Machinery (2019).
- [24] Wu, Y., Deng, L., Li, G., Zhu, J., and Shi, L., “Spatio-Temporal Backpropagation for Training High-Performance Spiking Neural Networks,” *Frontiers in Neuroscience* **12** (2018).
- [25] Lee, J. H., Delbruck, T., and Pfeiffer, M., “Training Deep Spiking Neural Networks Using Backpropagation,” *Frontiers in Neuroscience* **10** (2016).
- [26] Thiele, J. C., Bichler, O., and Dupret, A., “Event-Based, Timescale Invariant Unsupervised Online Deep Learning With STDP,” *Frontiers in Computational Neuroscience* **12** (2018).
- [27] Bi, G. and Poo, M., “Synaptic Modifications in Cultured Hippocampal Neurons: Dependence on Spike Timing, Synaptic Strength, and Postsynaptic Cell Type,” *Journal of Neuroscience* **18**(24), 10464–10472 (1998).
- [28] Gallego, G., Delbruck, T., Orchard, G., Bartolozzi, C., Taba, B., Censi, A., Leutenegger, S., Davison, A. J., Conradt, J., Daniilidis, K., and Scaramuzza, D., “Event-based vision: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).
- [29] Posch, C., Serrano-Gotarredona, T., Linares-Barranco, B., and Delbruck, T., “Retinomorphic event-based vision sensors: Bioinspired cameras with spiking output,” *Proceedings of the IEEE* **102**(10), 1470–1484 (2014).
- [30] Brandli, C., Berner, R., Yang, M., Liu, S.-C., and Delbruck, T., “A 240×180 130 db 3 μ s latency global shutter spatiotemporal vision sensor,” *IEEE Journal of Solid-State Circuits* **49**(10), 2333–2341 (2014).
- [31] “DVS camera.” <https://inivation.com/dvs/>. Accessed: 01/06/2020.

- [32] Mueggler, E., Rebecq, H., Gallego, G., Delbruck, T., and Scaramuzza, D., “The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and slam,” *The International Journal of Robotics Research* **36**(2), 142–149 (2017).
- [33] “Prophesee.” <https://www.prophesee.ai/>. Accessed: 31/07/2020.
- [34] “SmartThings Vision.” <https://www.samsung.com/au/smart-home/smartthings-vision-u999/>. Accessed: 31/07/2020.
- [35] Imam, N. and Cleland, T. A., “Rapid online learning and robust recall in a neuromorphic olfactory circuit,” *Nature Machine Intelligence* **2**(3), 181–191 (2020).
- [36] Lee, W., Tan, Y., Yao, H., Li, S., See, H., Hon, M., Ng, K., Xiong, B., Ho, J. S., and Tee, B. C. K., “A neuro-inspired artificial peripheral nervous system for scalable electronic skins,” *Science Robotics* **4**(32) (2019).
- [37] Neckar, A., Fok, S., Benjamin, B. V., Stewart, T. C., Oza, N. N., Voelker, A. R., Eliasmith, C., Manohar, R., and Boahen, K., “Braindrop: A mixed-signal neuromorphic architecture with a dynamical systems-based programming model,” *Proceedings of the IEEE* **107**(1), 144–164 (2018).
- [38] Benjamin, B. V., Gao, P., McQuinn, E., Choudhary, S., Chandrasekaran, A. R., Bussat, J.-M., Alvarez-Icaza, R., Arthur, J. V., Merolla, P. A., and Boahen, K., “Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations,” *Proceedings of the IEEE* **102**(5), 699–716 (2014).
- [39] Schemmel, J., Briiderle, D., Griibl, A., Hock, M., Meier, K., and Millner, S., “A wafer-scale neuromorphic hardware system for large-scale neural modeling,” in [*Proceedings of 2010 IEEE International Symposium on Circuits and Systems*], 1947–1950, IEEE (2010).
- [40] Qiao, N., Mostafa, H., Corradi, F., Osswald, M., Stefanini, F., Sumislawska, D., and Indiveri, G., “A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128k synapses,” *Frontiers in neuroscience* **9**, 141 (2015).
- [41] Moradi, S., Qiao, N., Stefanini, F., and Indiveri, G., “A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (dynaps),” *IEEE transactions on biomedical circuits and systems* **12**(1), 106–122 (2017).
- [42] Merolla, P. A., Arthur, J. V., Alvarez-Icaza, R., Cassidy, A. S., Sawada, J., Akopyan, F., Jackson, B. L., Imam, N., Guo, C., Nakamura, Y., et al., “A million spiking-neuron integrated circuit with a scalable communication network and interface,” *Science* **345**(6197), 668–673 (2014).
- [43] Davies, M., Srinivasa, N., Lin, T., Chinya, G., Cao, Y., Choday, S. H., Dimou, G., Joshi, P., Imam, N., Jain, S., Liao, Y., Lin, C., Lines, A., Liu, R., Mathaiikutty, D., McCoy, S., Paul, A., Tse, J., Venkataramanan, G., Weng, Y., Wild, A., Yang, Y., and Wang, H., “Loihi: A neuromorphic manycore processor with on-chip learning,” *IEEE Micro* **38**(1), 82–99 (2018).
- [44] Frenkel, C., Lefebvre, M., Legat, J.-D., and Bol, D., “A 0.086-mm² 12.7-pj/sop 64k-synapse 256-neuron online-learning digital spiking neuromorphic processor in 28-nm cmos,” *IEEE transactions on biomedical circuits and systems* **13**(1), 145–158 (2018).
- [45] Furber, S. B., Galluppi, F., Temple, S., and Plana, L. A., “The spinnaker project,” *Proceedings of the IEEE* **102**(5), 652–665 (2014).
- [46] Bekolay, T., Bergstra, J., Hunsberger, E., DeWolf, T., Stewart, T. C., Rasmussen, D., Choo, X., Voelker, A., and Eliasmith, C., “Nengo: a python tool for building large-scale functional brain models,” *Frontiers in neuroinformatics* **7**, 48 (2014).
- [47] Blouw, P., Choo, X., Hunsberger, E., and Eliasmith, C., “Benchmarking keyword spotting efficiency on neuromorphic hardware,” in [*Proceedings of the 7th Annual Neuro-inspired Computational Elements Workshop*], 1–8 (2019).
- [48] Eliasmith, C., Stewart, T. C., Choo, X., Bekolay, T., DeWolf, T., Tang, Y., and Rasmussen, D., “A large-scale model of the functioning brain,” *science* **338**(6111), 1202–1205 (2012).
- [49] Kheradpisheh, S. R., Ganjtabesh, M., Thorpe, S. J., and Masquelier, T., “Stdp-based spiking deep convolutional neural networks for object recognition,” *Neural Networks* **99**, 56–67 (2018).
- [50] Kirkland, P., Di Caterina, G., Soraghan, J. J., Andreopoulos, Y., and Matich, G., “UAV detection: a STDP trained deep convolutional spiking neural network retina-neuromorphic approach,” in [*28th International Conference on Artificial Neural Networks ICANN2019*], 724–736, Springer (2019).

- [51] Kirkland, P., Di Caterina, G., Soraghan, J., and Matich, G., “SpikeSEG: Spiking segmentation via STDP saliency mapping,” in [*International Joint Conference on Neural Networks IJCNN*], (2020).
- [52] Orchard, G., Jayawant, A., Cohen, G. K., and Thakor, N., “Converting static image datasets to spiking neuromorphic datasets using saccades,” *Frontiers in neuroscience* **9**, 437 (2015).
- [53] Li Fei-Fei, Fergus, R., and Perona, P., “Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories,” in [*2004 Conference on Computer Vision and Pattern Recognition Workshop*], 178–178, IEEE (2018).
- [54] Kirkland, P., Di Caterina, G., Soraghan, J., and Matich, G., “Perception Understanding Action: Adding Understanding to the Perception Action Cycle with Spiking Segmentation,” *Frontiers in Neurorobotics* **14**, 69 (2020).