



**HAL**  
open science

# Event-driven nearshore and shoreline coastline detection on SpiNNaker neuromorphic hardware

Mazdak Fatahi, Pierre Boulet, Giulia D'angelo

► **To cite this version:**

Mazdak Fatahi, Pierre Boulet, Giulia D'angelo. Event-driven nearshore and shoreline coastline detection on SpiNNaker neuromorphic hardware. *Neuromorphic Computing and Engineering*, 2024, 4 (3), pp.034012. 10.1088/2634-4386/ad76d5 . hal-04699880

**HAL Id: hal-04699880**

**<https://hal.science/hal-04699880v1>**

Submitted on 17 Sep 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright

# Event-driven nearshore and shoreline coastline detection on SpiNNaker neuromorphic hardware

MAZDAK FATAHI \*

Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France  
Mazdak.Fatahi@univ-lille.fr

PIERRE BOULET

Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France  
Pierre.Boulet@univ-lille.fr

GIULIA D'ANGELO

Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague  
Italian Institute of Technology, Event-Driven Perception for Robotics, Genoa, Italy  
giulia.dangelo@fel.cvut.cz

## Abstract

Coastline detection is vital for coastal management, involving frequent observation and assessment to understand coastal dynamics and inform decisions on environmental protection. Continuous streaming of high-resolution images demands robust data processing and storage solutions to manage large datasets efficiently, posing challenges that require innovative solutions for real-time analysis and meaningful insights extraction. This work leverages low-latency event-based vision sensors coupled with neuromorphic hardware in an attempt to decrease a two-fold challenge, reducing the computational burden to  $\sim 0.375$  mW whilst obtaining a coastline detection map in as little as 20 ms. The proposed Spiking Neural Network (SNN) runs on the SpiNNaker neuromorphic platform using a total of 18040 neurons reaching 98.33 % accuracy. The model has been characterised and evaluated by computing the accuracy of Intersection over Union (IoU) scores over the ground truth of a real-world coastline dataset across different time windows. The system's robustness was further assessed by evaluating its ability to avoid coastline detection in non-coastline profiles and funny shapes, achieving a success rate of 97.3 %.

## 1 Introduction

Shoreline monitoring is a critical aspect of coastal management, involving the frequent observation and assessment of shoreline locations and changes over time. This process is essential for understanding the dynamic nature of coastal areas and for making informed decisions regarding environmental protection and resource management. Remote sensing and standardised monitoring protocols map, analyse and predict shoreline changes to avoid potential environmental damage as well as to anticipate future changes in coastal areas. In coastal monitoring, both direct on-site observations and remote sensing techniques play pivotal roles in comprehensively understanding shoreline dynamics. Direct monitoring entails physical presence and surveys along the shoreline, allowing for the collection of detailed data on erosion rates, sediment transport, vegetation health, and anthropogenic impacts. Conversely,

---

\*Corresponding Author

remote monitoring utilizes satellite imagery, aerial drones, and other remote sensing technologies to cover broader areas, monitor changes over time, and access hazardous or inaccessible locations [56].

In recent years, the use of drones has revolutionised the field of monitoring and inspection of landmarks. Drones equipped with advanced visual perception have demonstrated the potential to reduce costs, enhance safety, and improve efficiency significantly [62]. Drones are employed in various practical applications, ranging from surveillance [36], aerial mapping [38] and inspection [49] to package delivery [10] and agriculture, where they operate autonomously. Another practical use of drones is shoreline monitoring of the coasts, playing a crucial role in conducting detailed surveys and capturing high-resolution imagery. In certain scenarios, drones are employed solely for data recording without employing automation in flight control, particularly when operating at elevated altitudes. Subsequently, the recorded data is subjected to shoreline extraction processes conducted offline, utilising GIS tools to generate the shoreline contour [61, 34]. Nevertheless, drone automation has been implemented utilising frame-based cameras [6] training of a Convolutional Neural Network (CNN) for detecting shoreline features from raw camera images as well as the optical flow estimation of the detected coastline.

The continuous streaming of high-resolution images results in an extremely consuming approach [57], necessitating robust data processing capabilities and efficient storage solutions to manage the influx of large datasets, posing challenges that demand innovative solutions for real-time analysis and meaningful extraction of valuable insights.

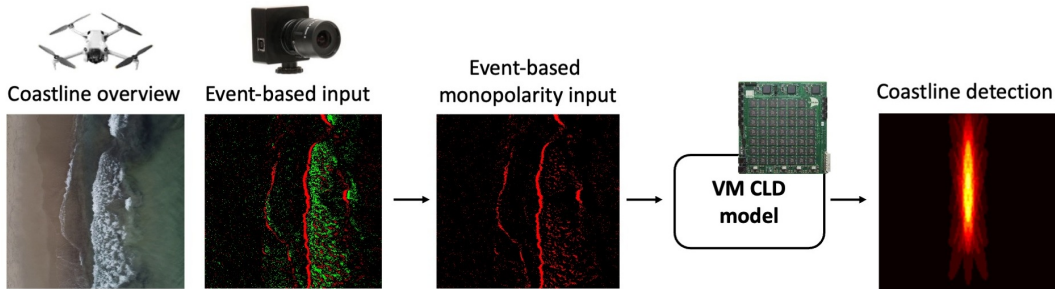


Figure 1: Overview of the bioinspired system. From the left to the right: the coastline overview from a drone, the event-camera output of the coastline scene with positive (red pixels) and negative polarities (green pixels), the monopolarity event-based input into the Von Mises Coast Line Detector (VM CLD) system running on the neuromorphic platform SpiNNaker, and the coastline detection map.

Recent approaches employ bioinspired event-based cameras and neuromorphic platforms, due to their efficacy in the reduction of latency and power consumption, disregarding redundant data while computing in parallel and contributing to the optimisation of resource utilisation. Event-driven cameras, inspired by biological processes, generate a sensor signal that loosely mimics mammalian visual responses and holds significant potential for integration with bioinspired vision processing [22]. These cameras produce "spikes" exclusively when and where there is a temporal contrast change [43], drawing inspiration from the mechanisms observed in mammalian vision [43]. This selective event generation results in a considerable reduction of data compared to conventional frame-based cameras operating at typical frame rates [43]. By eliminating redundant data streams, event-driven cameras substantially lower computational costs, boost processing speed and minimise latency [43]. These advantageous characteristics make event-based processing particularly suitable for robotics applications. Despite the prevalence of saliency-based approaches utilising Convolutional Neural Networks (CNNs) [25, 66], these methods often overlook the benefits of event-driven sparse computation crucial for robotic applications [37, 24], especially when employing fully spiking-based approaches [48, 59].

Recently, full bioinspired architectures have emerged for aerial vehicles. Specifically, for SNN online adaptation exploiting on-chip learning [59], efficient contour-based area tracking [8], and controlled freely flying drones [40]. Although these intelligent systems prove the efficiency of a fully bioinspired architecture, they do not specifically address the challenge of shoreline monitoring. This work aims to solve this challenge by leveraging event-based sensing in combination with neuromorphic computing to develop a low-power efficient system that significantly reduces latency and computational demands [55, 58]. The integration of the AER protocol with an SNN running on

the SpiNNaker [20] platform holds promising potential for enhancing the efficiency of the system opening the door to a new approach for coastline and nearshore coastline detection. To make the drone autonomous in near-shore missions and at low altitudes, we propose to use the near-shore moving waves which are mostly formed near the beach and where water touches the seabed ([60]).

The system is a follow-up and adjustment of a saliency-based visual attention model running on SpiNNaker [14] producing a response in  $\sim 16$  ms. Despite using the same kernel filter, the arrangement of the model is novel and tailored for shoreline monitoring. The proposed method lies on event-driven cameras in conjunction with an SNN and neuromorphic platform for detecting and localising shorelines. This integration enhances the efficiency of the overall system, fostering seamless communication and collaboration between the bioinspired algorithms implemented in software and the specialised hardware designed to process the event-driven signals. The high dynamic range exhibited by event-driven cameras further improves their utility for nocturnal operations [63, 30]. This unique characteristic extends the model’s applicability to nighttime monitoring scenarios, enabling continuous and comprehensive observations even during low-light conditions. The contribution of the paper lies in the following:

- Characterisation of the model across different time windows,  $\Delta T$ s.
- Comparison of the model to real-world data against ground truth using IoU scores.
- Accuracy of the model performances.
- Robustness evaluation over non-coastline profiles.

The model effectively detects shoreline moving waves and introduces a novel application, specifically, the identification of the swash edge line, encompassing both uprush (the movement of water up a beach or shore during the advance of a wave) and backwash (the movement of water back toward the ocean or body of water after a wave has broken) [11]. The full bioinspired system opens a new approach to the investigation and monitoring of dynamic coastal phenomena, whilst reducing the latency and the power consumption required in robotics, especially using drones.

## 2 Nearshore waves and coastal environmental monitoring

In the realm of coastal environmental monitoring, the detection and understanding of wave dynamics are essential across a spectrum of investigations. These investigations span from addressing issues such as beach waste and litter detection and localisation [3, 4, 5] to more complex studies like coastal flooding prediction, monitoring coastal erosion, and examining the coastal response to storms [32, 12, 1, 11, 29, 17, 44].

The implementation of remote survey methodologies signifies a notable leap forward compared to conventional on-site survey methods, particularly in nearshore regions. This advancement allows for more efficient data collection and analysis without the limitations posed by physical presence, thereby enhancing our understanding and management of coastal environments. Techniques employing Light Detection and Ranging (LiDAR) scanning [26, 35], satellite images [42, 27], and aerial images captured at different altitudes (on-line and off-line)[17, 7] have become prominent[65]. These methods not only streamline the survey process but also enhance the efficiency of data collection in coastal monitoring initiatives.

The mentioned approaches can become very expensive or rely heavily on the quality of the images used, therefore, some of the latest novel approaches exploit the neuromorphic approach in an attempt to reduce latency and power consumption. Within the domain of path and trajectory tracking utilising SNNs and neuromorphic hardware, significant advancements have been made as evidenced by the works of Paredes-Vallés et al. [40]. This work is a comprehensive exploration conducted on the development of a fully neuromorphic vision-to-control pipeline tailored for autonomous drone flight. Specifically, the investigation prioritises the utilisation of the Loihi [15] neuromorphic processor, showcasing its capabilities in this domain. Meanwhile, Vitale et al. [59] delve into tracking applications, particularly in the context of unmanned aerial vehicles (UAVs), by integrating an SNN for vision-based control. This study employs an event-based camera coupled with the Hough transform algorithm, demonstrating the feasibility of implementing the SNN controller on Loihi [15]. Moreover, it highlights the potential for high-speed control tasks, particularly in horizon tracking, when compared to conventional event-driven CPU controllers. Additionally, the Event-Based Tracking Control Framework from Aspragkathos et al. [8] presents an event-based control framework tailored for multirotor aerial vehicles (MAVs), emphasising efficient tracking of contour-based

areas like road pavements. This framework employs a bio-inspired Dynamic Vision Sensor (DVS) [50] and an SNN running on the SpiNNaker (SpiNN-3 board), effectively processing camera data and maintaining tracked areas within the DVS camera’s field of view. The study introduces an event-based Neuromorphic Hough Transform to facilitate SNN execution on neuromorphic chips, aiming to extract pertinent lines from the visual scene using events from the DVS. These works collectively underscore the promising potential of SNNs and neuromorphic hardware in advancing path and trajectory-tracking applications in various unmanned systems.

However, none of the referenced works specifically address the detection of nearshore waves to maintain a drone on the desired trajectory for tracking the nearshore region. To the best of the authors’ knowledge, there is currently no existing literature on event-based neuromorphic implementations in this specific application domain, highlighting a significant gap in the research landscape.

To achieve this objective, this work implements an SNN taking inspiration from the visual attention model proposed by D’Angelo et al. [14], deploying it onto the SpiNNaker neuromorphic board and leveraging a biologically plausible event-based architecture for object detection in visual attention [28, 23].

### 3 From visual attention to coastline detection

Visual attention refers to an agent’s ability to focus only on relevant parts of the scene, thus dramatically reducing the amount of information to be processed. Recently, various visual attention event-based and spiking-based architectures have been proposed in robotics, demonstrating the capability to achieve a fast ( $\sim 100\text{ms}$ ) and reliable response for attention exploiting cues such as intensity [28] and depth information [23]. These approaches rely on the Von Mises (VM) distribution, which can detect close contours based on Gestalt theories of perceptual grouping, introducing the importance of **border ownership perception** in attention. The information of the detected closed contours is then pooled from a Grouping Cells layer to look for areas where potential objects can be found, also called "proto-objects". The VM distribution emulates the border ownership cells in the Secondary Visual cortex V2 [64]; these cells in the visual cortex are responsible for the perception of which side of a border (edge) "owns" the border, i.e., whether it belongs to a figure or its background, therefore distinguishing the foreground from the background. This curved kernel ("half moon" shape) was initially introduced in the context of visual attention by Russell et al. [47], encompassing various orientations and scales. Combining two opposing curved VM filters across different orientations enables the detection of potential objects, thereby delineating salient areas within the visual field, akin to proto-objects. The model has been further modified to event-based and spiking-based approaches to enhance the applicability of robotics applications running online on the humanoid robot iCub with low latency and power consumption [28, 14, 23].

The proposed approach is a follow-up of the SNN saliency-based visual attention model [14] exploiting the inherent characteristics of the VM kernel shape. The idea is to use the VM kernel as it resembles the curved shape of the waves. The previous work [14] is based on two fundamental layers called Border Ownership and Grouping Pyramids, where opposite orientations of VM filters (i.e.,  $0$  and  $0 + \pi$ ) are connected together to enhance the presence of close contours, such as proto-objects. The Border Ownership layer detects curved edges of different orientations, and the Grouping layer pools the information, enhancing the presence of potential objects in the scene (producing a saliency map roughly every  $\sim 16\text{ms}$ ). The model is based on the Gestalt rules of perceptual grouping, which found evidence in V2 [64]. In the proposed case for coastline detection, the only similarity is the choice to use the VM kernel and to split the kernel into 4 segments to achieve accurate detection of curved shapes. We chose this kernel for its versatility in detecting both vertical and curved lines. In our case, the orientations we considered are only  $[-15^\circ, 0^\circ, +15^\circ]$ , to detect orientations of the coastlines slightly different from the perfectly vertical across the visual field. Since the target of this work is not object detection based on close contours, there is no connection between VM filters of opposite orientations (as in the visual attention model [14]) because it would be counterproductive for coastline detection. Therefore, the network architecture has been completely restructured. The absence of SNN models for coastline detection on neuromorphic platforms, coupled with the remarkable latency results achieved in [14], and the compatibility of the VM kernel shape with waveforms, prompted us to pursue this particular approach.

## 4 Methods

The entire system is designed to support both static and dynamic monitoring, including the possibility of running on a drone at a consistent altitude and speed (see Figure 1). The complete bioinspired architecture processes events from an event-based bioinspired camera as input and feeds them into the system implementation running in simulation on the neuromorphic platform SpiNNaker. [20] (see Figure 1). The target neuromorphic hardware limits the number of emulated neurons per core, which varies based on the complexity of the neuron model used [20]. To ensure the model can run on SpiNN-3, the system is designed to fit on a drone, and the implementation aims to optimise and minimise the model size and the number of required neurons. Similarly, the visual input has been cropped to  $128 \times 128$  pixels, so only events from the cropped centre are processed.

### 4.1 Von Mises Coast Line Detector (VM CLD): Network architecture

The Von Mises Coastline Detector (VM CLD) includes four processing populations that split the architecture into sub-blocks: the von Mises layers, the Filter Segments Neurons, the Filter Neurons and the Vertical Denoising Neurons (see Figure 2). Described in detail as follows:

- *VM layers*: The first layer is composed of 9 layers of VM filters covering the entire visual field with three orientations ( $\theta = [-15^\circ, 0^\circ, +15^\circ]$ ) and three scales to ensure scale invariance. Due to the Integrate-and-Fire with Current-based synapses and Exponential decay behaviour of each neuron, random noise does not contribute to increases in membrane potential [13].
- *Filter Segments Population*: This population ensures the correct curved-shaped detection of the VM filter. Therefore, each VM filter is represented by 4 neurons, splitting the VM shape into four sections and ensuring activation from that area only if all the segments are active and close in time. The population has 32 sets of 4 neurons each, covering the entire visual field, resulting in 128 neurons per layer (See Table 1).
- *Filter Neurons Population*: This population receives excitatory connections from the segments of the VM Filter Segments Population. It effectively combines each group of four neurons into a single Filter Neuron. Additionally, the population receives inhibitory input from the background part of the VM filter (black part in Figure 2), enhancing the selection for curved-shaped activity.
- *Vertical Denoising Population*: This additional population has been implemented solely to focus the model on relevant vertical active regions, thereby enhancing the system’s selectivity for coastline detection. It achieves this by vertically pooling every two adjacent neurons from the previous layer across the visual field.

The complete architecture incorporates the minimum orientations and scales to ensure correct coastline detection and scale invariance. The output of the VM CLD model is the CLD map, built by collecting the output spikes from the network over a period of time.

#### 4.1.1 Filter Segments Population

The system takes as input the stream of events and processes them through the *Von Mises layers* which are the fundamental kernels used to detect the coastline, described in Equation 1 (see Figure 3).

$$VM_{\theta}(x, y) = \frac{\exp(\rho \cdot R_0 \cdot \cos(\text{atan2}(-Y, X) - \theta))}{I_0(\sqrt{X^2 + Y^2} - R_0)} \quad (1)$$

Where  $x$  and  $y$  are the kernel coordinates with the origin in the top-left corner of the filter with a size equal to  $M \times M$ ,  $X$  and  $Y$  are the kernel coordinates with the origin in the centre of the filter ( $X = x - M/2, Y = y - M/2$ ),  $R_0$  is the radius of the filter (the position of the curved filter relative to the kernel centre),  $\theta$  the orientation of the filter,  $I_0$  stands for the modified Bessel Function of the first kind, and  $\rho$ , which governs the arc length of active pixels within the kernel, thereby influencing the convexity of the kernel. For  $\rho < 1$ , the filter exhibits increased sensitivity towards convex shapes as opposed to straight lines, rendering it more suitable for coastline detection tasks. Empirically, we have determined that a value of  $\rho = 0.2$  effectively detects convex shapes while simultaneously

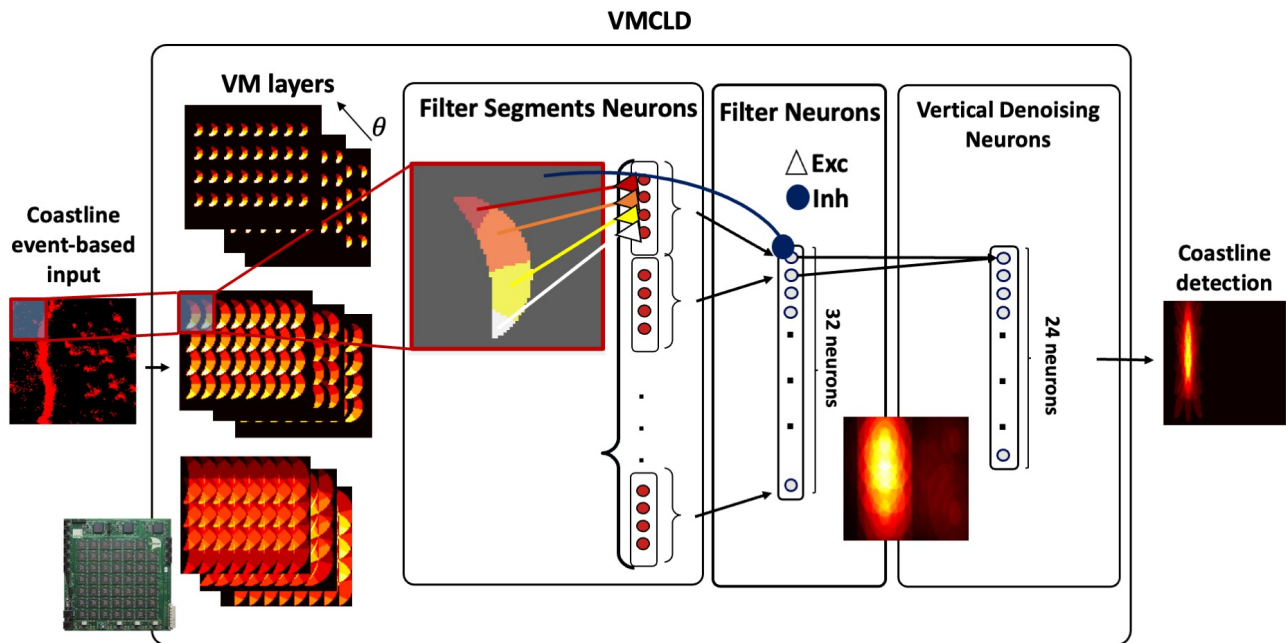


Figure 2: Overview of the VM CLD architecture running on the SpiNNaker 3 neuromorphic platform. From left to right: coastline event-based input, VM layers (three orientations  $\theta = [-15^\circ, 0^\circ, +15^\circ]$  and three scales), Filter Segments Neurons (four segments neurons for each kernel: red, orange, yellow and white section), Filter Neurons (Integrating the outputs of the four corresponding neurons from the Filter Segment Neurons), Vertical Denoising Neurons (minimising interference from non-vertical active regions) and the final Coastline Detection (CLD) map.

filtering out straight lines. The VM filters cover the visual field with a horizontal and vertical stride of 24 pixels. The VM layers orientations employed in our approach are  $\theta = [-15^\circ, 0^\circ, +15^\circ]$  to ensure detection of slightly oriented curved coastlines. By ranging  $(x, y)$  from 0 to the size of the filter ( $M$ ),  $VM_\theta(x, y)$  provides the pixel’s value for each given pixel  $(x, y)$ . This results in the desired filter with a scale of  $M * M$ , an orientation of  $\theta$  and a radius of  $R_0$  (see Figure 3).

To enhance the selectivity of the VM filter, the values inside and outside the curved shape are set as follows:

$$\hat{VM}_\theta(x, y) = \begin{cases} +1 & \text{for } VM_\theta(x, y) > 0.75 \\ -1 & \text{else} \end{cases} \quad (2)$$

This binarisation of the VM filters to +1 or -1 enhances the selectivity of the kernel to curved shapes. Where  $\hat{VM}_\theta(x, y)$  represents the filter with the binary values. Since coastline detection does not require distinguishing between positive and negative events during processing, for simplicity, only events with positive polarity will be considered and input into the model. Furthermore, events with positive polarity tend to form in front of the waves and clearly indicate their direction of movement (see Figure 1). This deliberate choice serves a dual purpose: firstly, it reduces the network size and the active neuron count (scaled down by a factor of 2), and secondly, it mitigates computation costs, subsequently leading to a reduction in energy consumption (see Figure 1).

The first computational layers aim to extract the region of interest, namely possible areas containing the coastlines, from the Field of View (FoV) across different scales and orientations. The convolution with multi-scale filters ensures scale invariance to allow a range of possible altitudes for the monitoring system. Taking into account the assumptions that coastlines will be always aligned and continuous, the angles  $-15^\circ, 0^\circ$ , and  $+15^\circ$  are employed (Figure 3).

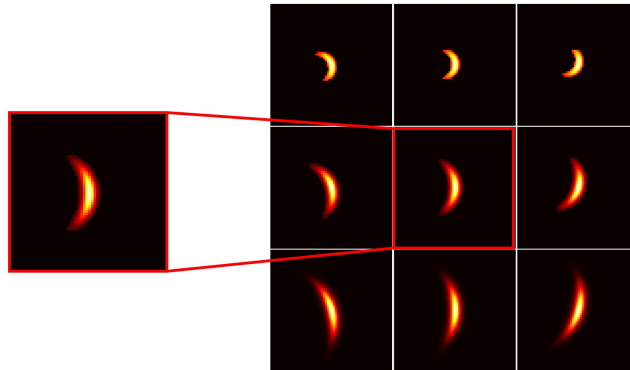


Figure 3: Overview of Von Mises kernels. On the left: VM kernel. On the right: Three different sizes ([8, 16, 32] pixels) and orientations ( $[-15^\circ, 0^\circ, +15^\circ]$ ) used in the VM CLD model.

Each VM filter connects every neuron to the *Filter Segments population*. This process involves mapping the pixels within each filter’s receptive field to the corresponding neurons in the Filter Segments population through the connections implementing convolutional operations.

To mitigate the impact of ‘hot’ pixels [33] and small noisy regions characterised by a high occurrence of irrelevant events, each VM filter is subdivided into four rotationally distributed segments. [14]. To ensure equal activation of each segment, the corresponding neuron in the Filter Segment layer adjusts the weights for equal contribution from all the segments. This approach ensures a fair distribution of influence across all segments, promoting balanced triggering of neurons within the Filter Segments population. Consequently, every pixel within the receptive field of each segment in every VM filter is directed towards a specific neuron within the Filter Segments population. In other words, four neurons within the Filter Segments population collectively respond to a single VM filter.

As a result, each Filter Segments population in every VM CLD contains 128 neurons.



### 4.1.2 Filter Neurons Population

The model’s exclusive responsiveness to input events confined within the VM kernel is ensured by connecting only the segments to the corresponding neuron in the Filter Neuron population with excitatory synapses. Additionally, connection weights outside the segments are linked to the corresponding neuron in the Filter Neuron population with inhibitory synapses. Consequently, spikes occurring outside the designated filter area, excluding those from the VM kernel, exert an inhibitory influence on the aggregate filter neuron, thereby enhancing its sensitivity to the VM shape. This approach to selecting the VM filter shape is achieved through inhibitory connections. Considering the targeted neuromorphic hardware (SpiNNaker), which lacks negative weights, these inhibitory connections function similarly to the negative weights in kernels for traditional convolution. This approach helps to disregard false coastlines caused by waves on the surface of the water. As a result, the model becomes extremely sensitive to the interested shapes (curved moving waves) while disregarding irrelevant activities or regions, thereby ensuring a focused response to pertinent inputs and significantly to the regions containing the edge of the moving waves. To further diminish the model’s sensitivity to irrelevant active regions, an additional fine-tuning of the neural parameters, particularly the threshold and the refractory period, for the corresponding neuron within the Filter Neurons population has been performed (see Table 2). The fine-tuning ensures the Filter Neurons population spikes only when all four inputs are simultaneously firing, aligning with the goal of refining the model’s responsiveness to relevant signals while disregarding irrelevant activation.

### 4.1.3 Vertical Denoising Population

While we have developed a pyramid of VM filters to ensure scale invariance, another focus of the VM CLD model’s approach is to reduce the occurrence of relatively large waves forming parallel to the shoreline. Despite the reduction in active pixels achieved by eliminating negative polarity, numerous irrelevant and expansive active regions persist, as shown in Figure 4. These areas, though fewer, can contribute to numerous distracting active zones in the final output, thereby complicating the detection of the precise location of continuous waves.

To address this issue, the last step of the VM CLD model implements a mechanism in which each vertical pair from the Filter Neuron population is integrated under the assumption that the coastline is consistently vertical through the whole visual field (Figure 4).

The neurons within the Vertical Denoising Population fire only when two consequent neurons part of a vertical segment of the moving wave are concurrently active. This approach enhances the precision of wave detection by ensuring that the output reflects the firing patterns of neuron pairs corresponding to the actual progression of the waves, thereby minimising interference from irrelevant or distracting regions. Due to the vertical grouping, the ultimate layer exhibits a decrease in the neuron count, from 32 neurons in the Filter Neuron Population to 24 neurons in the Vertical Denoising Population. This approach mitigates the impact of small, undesired active regions while enhancing coastline detection (see Figure 4).

## 4.2 Model deployment: SpiNNaker neuromorphic platform

SpiNNaker [20], one of the pioneering neuromorphic platforms [9], was designed to fulfil the needs of low power consumption and low latency through parallel computing for SNNs. This platform is distinctive for its biologically inspired, massively parallel computing system tailored specifically for modelling and simulating large-scale real-time networks. The proposed architecture exploits the SpiNN-5 (SpiNNaker 103) board [39, 21], comprising 48 SpiNNaker chips. Within each SpiNNaker chip, there are 18 ARM cores, each equipped with a 32kB ITCM (instruction tightly coupled memory) and a 64kB DTCM (data tightly coupled memory). Additionally, a 128 MB SDRAM is collectively shared among the 18 cores. To emulate the intricate connectivity observed in the brain, these cores are interlinked through an asynchronous Network-on-Chip (NoC) utilising a multicast packet-routing mechanism. Furthermore, SpiNN-5 incorporates three Xilinx Spartan-6 FPGAs to facilitate high-speed serial links. Facilitating the connection between the SpiNNaker board and the computer is a 100 MB Ethernet controller. This controller serves as the conduit for loading data into the SpiNNaker memory, enabling the execution of real-time simulations. Moreover, sPyNNaker [45] is a software package employed to define models using the PyNN script [16]. The proposed network, consisting of the 9 parallel populations, is implemented as a PyNN model, which plays a crucial role in translating these models into a format suitable for execution on the SpiNNaker platform. One of the

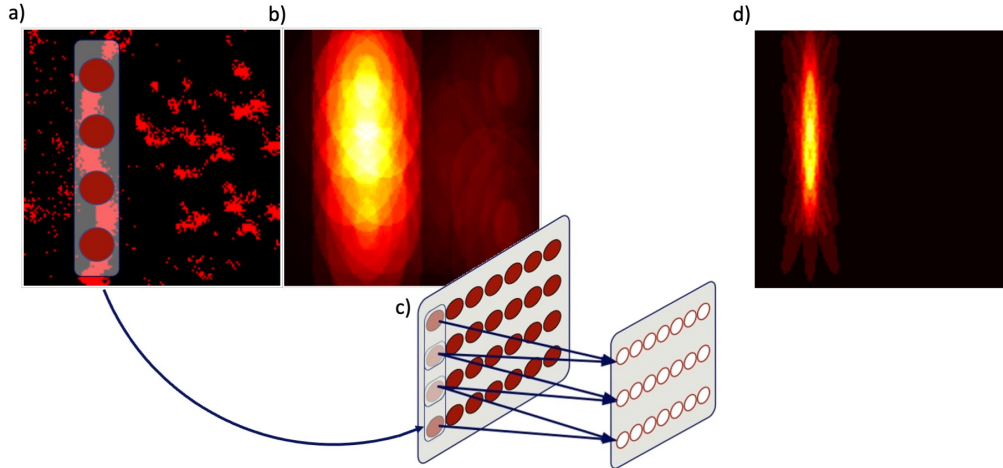


Figure 4: From the left to the right: a) Overview of the positive events collected in a period of 50 ms showing the challenge in accurately locating the desired waves defining the coastline. Overlapped over the event frame the representation of the vertical grouping on the Filter Neurons layer. b) Response of the VM CLD model without the Vertical Denoising Population. c) Representation of the Filter Neurons projecting to the Vertical Denoising Population. The approach eliminates small and irrelevant active areas by integrating vertical pairs of neurons from the Filter Neuron population into the Vertical Denoising Population. d) Response of the system after the vertical denoising approach.

significant constraints in deploying spiking models on the SpiNNaker board relates to the limitation on the number of supported neurons. The theoretical limits for the SpiNN-5 board are approximately  $\sim 220k$ , and for the SpiNN-3 board, it is around  $\sim 18k$  [19]. Due to these constraints, the VM CLD model defines a resolution of 128 by 128, resulting in 32,768 neurons only for the first layer representing the visual field. Despite using the SpiNN-5 board in this work, the physical size and weight of the board may not be suitable for the real-world application at hand, such as shoreline surveillance using Unmanned Aerial Vehicles (UAVs). Therefore, although the model is tested and developed on the SpiNN-5 board, the model is constrained to meet the limitations of the SpiNN-3 board. By only processing positive events, the number of neurons in the first layer can be halved, aiding in the deployment of the model on the SpiNN-3 board.

Table 1 illustrates the number of required neurons in each layer. Except for the input layer (16384 neurons), which is common across all 9 pipelines (3 orientations and 3 scales), requires 184 neurons (128 Filter Segments Neurons Population + 32 Filter Neurons Population + 24 Vertical Denoising Population). The total model’s requirement of 18,040 neurons finally appears to be well-suited to the constraints of the SpiNN-3 platform. Table 2 shows the parameters used for the network’s neurons. The system employs *IF\_curr\_exp* neurons, referring to “Integrate-and-Fire with Current-based synapses and Exponential decay” neurons implementable in PyNN, the *exp\_curr*.

#### 4.2.1 Model Latency

The discussion on the latency of the system holds a critical aspect for real-time applications such as online drone tracking. The proposed architecture can be split into three key checkpoints, each accompanied by its respective delays:

- *Interfacing DVS to SpiNNaker board*: This latency is intricately tied to the system integration of hardware. The integration latency is strictly contingent upon custom configurations, such as the DVS-SpiNNaker interface type and the drone’s onboard computer. The proposed method involves utilising FPGA implementation

Table 1: Number of the neurons per population.

Populations	No. Neurons
<b>Input Population</b>	16384 (128×128)
<b>Filter Segments Neurons Population</b>	128 (32×4)
<b>Filter Neurons Population</b>	32
<b>Vertical Denoising Population</b>	24
<b>Total</b>	16384 + 184 * 9 = 18040

Table 2: Network parameters for each layer.

Layer	$\tau_m(ms)$	$v_{th}(mv)$	$\tau_{ref}(ms)$	$v_{rest}(mv)$	$v_{reset}(mv)$	$\tau_{syn_E}(ms)$	$\tau_{syn_I}(ms)$	$I_{offset}(mA)$
Filter Segments Neurons	20.0	-50.0	0.1	-65.0	-65.0	5.0	5.0	0.0
Filter Neurons	20.0	-40.0	0.12	-65.0	-65.0	5.0	5.0	0.0
Vertical Filter Neurons	20.0	-20.0	0.1	-65.0	-65.0	5.0	5.0	0.0

for interfacing the DVS camera with the SpiNNaker board, which, according to literature [41, 6], imposes a latency lower than 1 microsecond.

- *Model running on SpiNNaker*: This latency represents the time required for an event to be processed by the model on the SpiNNaker board, essentially constituting the processing time needed to generate output spikes. Considering the proposed architecture and PyNN implementation, all synaptic delays are set to 1 millisecond and there are three synaptic connection sets:
  - Events input → Filter Segments Neurons
  - Filter Segments Neurons → Filter-Neurons
  - Filter-Neurons → Vertical-Denoising-Neurons

Therefore, each spike necessitates only 3 milliseconds to become available at the output.

- *Model on CPU*: This encompasses the time required for framing the output events (accumulating the output spikes), identifying the largest contour as the region of interest, and issuing velocity commands.

The negligible latency of the integration, coupled with a mere 3 ms requirement for generating initial spikes at the output, and adopting, for example,  $\Delta T = 40$  ms for the collection of events estimate a total latency of less than 50 ms, thus facilitating real-time responsiveness and efficacy in real-world wave-tracking applications.

## Code and Data Availability

The code used for this work is available on GitHub at <https://github.com/MazdakFatahi/EventBased-VM-CoastLine-Detector>. The data is available at <https://nextcloud.univ-lille.fr/index.php/s/KeJtra48NTkemPM>.

## 5 Experiments and Results

Experiments and ground truth generation have been carried out thanks to data collected from the APROVIS3D project [2] captured near UCA in Antibes using a DJI Mini drone, employed with altitudes ranging from 20.2 to 20.8 meters moving at a speed of 0.28 m/s<sup>1</sup>. In this dataset, named the "Antibes dataset", onshore waves propagate towards the coastline and interact with the seabed upon reaching it.

To quantitatively assess the performances of the system, the model has been characterised and evaluated through different experiments, as follows:

<sup>1</sup>Further video showing the VM CLD response at [https://youtu.be/uL13H0\\_BBx8](https://youtu.be/uL13H0_BBx8).

- Characterisation of the system latency figures.
- Identification of metrics and ground truth to evaluate the performances of the model.
- Evaluation of the model, exploiting IoU scores, across varied  $\Delta T$  intervals of events collection.
- Further evaluation of the performance through IoU thresholding.
- Assessment of the model robustness.

## 5.1 Metrics & Ground Truth Generation

The validation has been carried out to assess the effectiveness of the model in detecting locations containing moving waves. The chosen **metric** for evaluation is the IoU, also known as the Jaccard Index, which is a widely recognised metric in image segmentation across a variety of applications [31, 51, 52]. The IoU (equation 3) serves as a critical metric that quantifies the ratio of overlap between the pixels predicted by a model and the ground truth pixels within an image. Essentially, the IoU provides a valuable evaluation of the model’s performance, specifically focusing on the pixel-level alignment between predictions and ground truth.

$$\text{IoU}_i = \frac{\text{Area}(\text{BBOX}_{\text{Pred}_i}) \cap \text{Area}(\text{BBOX}_{\text{GT}_i})}{\text{Area}(\text{BBOX}_{\text{Pred}_i}) \cup \text{Area}(\text{BBOX}_{\text{GT}_i})} \quad (3)$$

Where  $\text{Area}(\text{BBOX}_{\text{Pred}_i})$  represents the area of the predicted (detected) bounding box surrounding the region of interest in the  $i^{\text{th}}$  output and  $\text{Area}(\text{BBOX}_{\text{GT}_i})$  is the area of the  $i^{\text{th}}$  ground truth bounding box. To obtain a binary segmentation distinguishing between detected coastlines and non-coastlines, a binary classifier is utilized to determine whether the predicted region includes the Region of Interest (RoI) or not.

To perform the evaluation, we required a dataset to compare the detected areas with the **ground-truth**. The conversion of the RGB data from the Antibes dataset into events has been obtained by playing the video on an HD LED monitor (HP P24q G4) and recording with an event-based Prophesee HD camera (EVK4 HD [18]). Subsequently, only the events coming from a central region of  $128 \times 128$  have been collected.

To investigate the latency of the system’s response, the outgoing spikes from the VM CLD model were collected over various time windows, and compared to the ground truth. After the collection of the events into frames, a meticulous manual annotation procedure generating bonding boxes encapsulating the RoIs ( $\sim 5000$  images) has been performed to obtain the final ground truth.

## 5.2 IoU Across Varied Time Windows

Given that the model is intended for shoreline surveillance tasks using drones, it is imperative to send control commands to the drone’s flight unit at the appropriate time to maintain the drone on the desired trajectory. Therefore, the time the model waits to accumulate output spikes directly affects the system’s response time. For this reason, the evaluation of the model’s performance has been conducted across various time windows,  $\Delta T$ , with values [20, 30, 40, 50] ms, as this parameter is crucial. The experiment has been conducted on the Antibes dataset, 22 seconds long, therefore over 440 event frames for a  $\Delta T$  of 50ms. To determine the optimal  $\Delta T$  crucial for optimising the model’s performance, Table 3 summarises the average IoU scores.

Table 3: Percentage of IoU scores in average over different  $\Delta T$ s across all instances for a specific time window. The IoU score, ranging between 0 and 1, reflects the degree of overlap between the output spikes and the ground truth. A score of 0 signifies a complete lack of overlap, while a score of 1 indicates a perfect match with 100 % overlap.

$\Delta T$ [ms]	Average of IoU [%]
20	61.68
30	65.66
40	67.06
50	<b>69.59</b>

From Table 3, due to the increased number of events collected, there is a better definition of the coastline on the event frame. It becomes evident that larger values of  $\Delta T$ s (40 ms & 50 ms) contribute to superior results, 67.06 %

& 69.59 % respectively. To gain a comprehensive understanding of the individual contributions shaping the overall average, Figure 5 shows the distribution of IoU scores across different  $\Delta T$ .

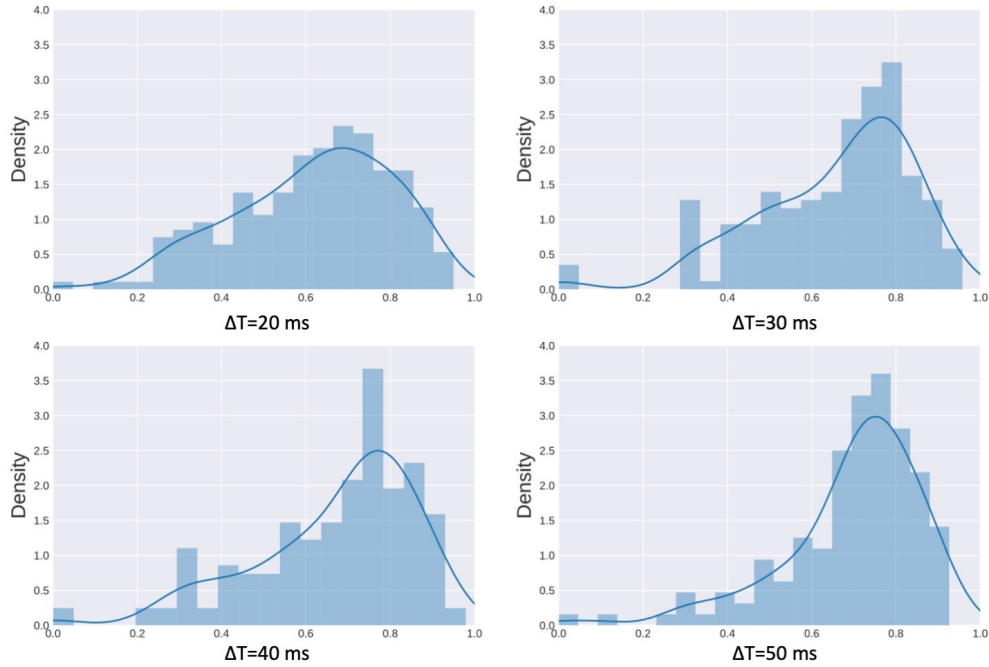


Figure 5: Overview of IoU scores over different  $\Delta T$ . Density distribution of the IoU scores ( $y$  axis) for different  $\Delta T$ s [20, 30, 40, 50] ms (IoU scores,  $x$  axis).

From Figure 5, it is evident that the concentration of IoU scores exceeding 0.5 is predominantly on the right side of each graph, indicating a high density of high IoU scores and thereby confirming their frequent occurrence from the system. Despite the high density for high IoU scores, Figure 6 shows the system response with low IoU scores, demonstrating that the model remains capable of detecting waves.

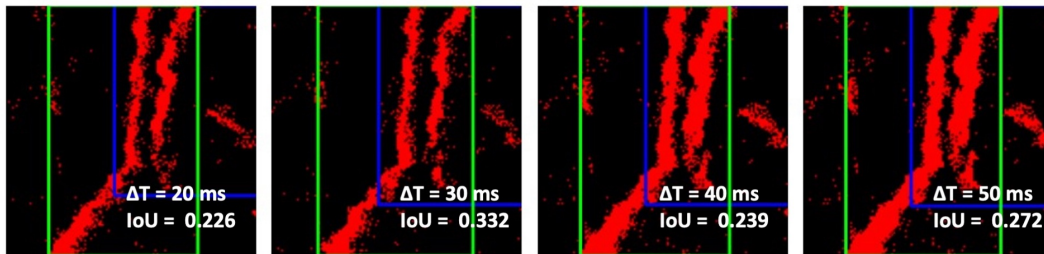


Figure 6: Examples of event frames with bounding boxes of the model for wave detection (blue) and ground truth (green) across  $\Delta T$ s, along with the corresponding IoU scores. These examples show correct wave detection despite the low IoU scores.

### 5.3 Model as a Binary Classifier: Evaluating Performance through IoU Thresholding

To frame the problem as a binary classification, whether the wave is detected or not, and consequently assess the accuracy, IoU scores are treated as logits by applying a threshold. IoU values below the threshold are designated

as class 0 (False, the wave is not detected), while those above the threshold are classified as class 1 (True, the wave is well detected) (See Eq. 4).

$$\text{Prediction} = \begin{cases} 1 & \text{if IoU} \geq \text{threshold} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Table 4 shows the impact of employing various threshold values across different  $\Delta T$ . The higher the threshold, the lower the selectivity of detection since the under-curve area increases; therefore, a greater number of scores are categorised as class 1. The maximum accuracy across all experiments is achieved when applying a threshold equal to 0.30. However, an interesting observation from Table 4 is that employing such a threshold yields the highest score for  $\Delta T=30$  ms. This finding is somewhat unexpected, as one might have anticipated a higher score for  $\Delta T=50$  ms, thereby increasing the likelihood of detecting a coastline due to the greater number of events collected. In conclusion, it should be noted that prioritising selectivity (with a threshold of 0.40) and looking for the minimum  $\Delta T$  (40 ms) with high accuracy (89.82 %), would indicate the optimal selection for a low-latency system (refer to Table 4).

Table 4: Model accuracy across different IoU score thresholds (from most to least selective) and  $\Delta T$  values.

$\Delta T$	Average of IoU	Accuracy [%] (Threshold = 0.80)	Accuracy [%] (Threshold = 0.70)	Accuracy [%] (Threshold = 0.60)	Accuracy [%] (Threshold = 0.50)	Accuracy [%] (Threshold = 0.40)	Accuracy [%] (Threshold = 0.30)
20	61.68	18.69	37.88	57.07	73.23	85.35	92.93
30	65.66	22.78	49.44	67.78	78.89	90.56	<b>98.33</b>
40	67.06	<b>26.35</b>	55.69	70.66	82.04	89.82	95.81
50	<b>69.59</b>	24.64	<b>57.97</b>	<b>78.99</b>	<b>87.68</b>	<b>93.48</b>	96.38

To understand the responses depending on the different thresholds further investigation has been explored on the distribution of the IoU scores. Figure 7 shows an increasing selectivity to high IoU scores for each threshold increasing the  $\Delta T$ . These results align with the hypothesis that the more events are collected, the higher the chance of detecting the wave correctly. Figure 7 also indicates that for time windows ( $\Delta T$ ) of 40 and 50 ms, a threshold higher than 0.5 still encompasses the majority of the IoU score density.

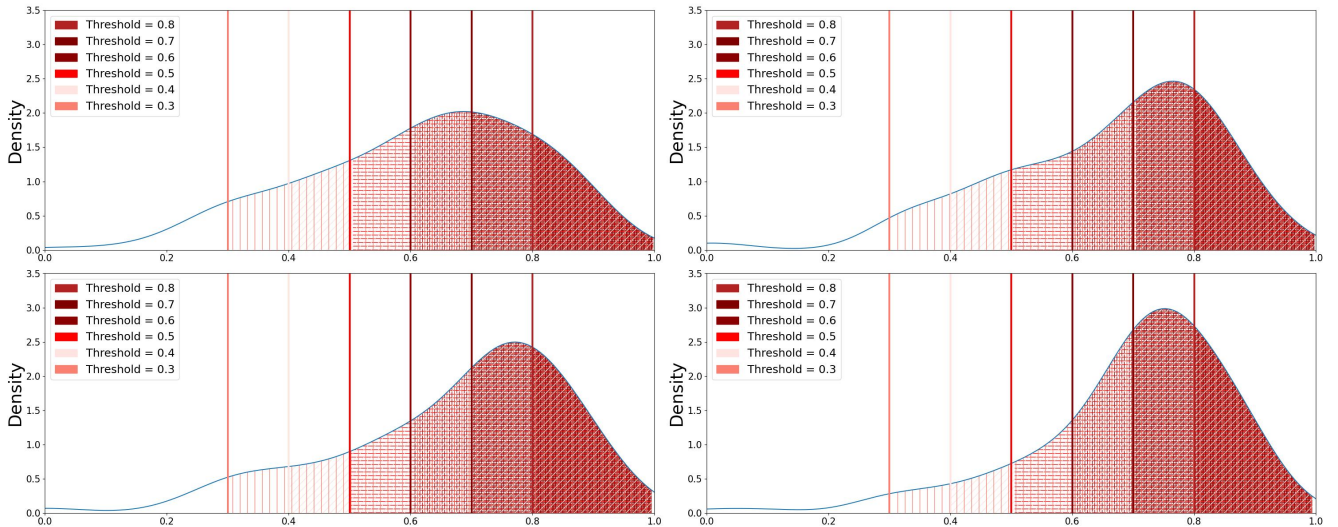


Figure 7: Distribution of IoU scores across different  $\Delta T$ s [20, 30, 40, 50] ms (from the top-left to the bottom-right) and thresholds [0.3, 0.4, 0.5, 0.6, 0.7, 0.8], reporting on the  $x$  axis the IoU scores and the Density distribution of the IoU scores on the  $y$  axis.

In conclusion, increasing  $\Delta T$  while simultaneously decreasing the threshold can result in higher accuracy rates.

However, it’s important to note that elongating  $\Delta T$  implies a longer accumulating period of the output spikes, prolonging the decision-making to maintain the drone on its trajectory. Conversely, increasing the threshold diminishes the percentage of acceptable overlap considered as the True class. As this parameter delineates the condition of acceptable detection, it is intricately linked to the application’s requirements and objectives. Therefore, striking a balance between  $\Delta T$ , threshold settings, and the operational context is crucial for optimising performance and ensuring the effectiveness of trajectory tracking in practical scenarios.

## 5.4 Model Robustness: Navigating Stability of the model

While this study delves into the efficiency of the model in terms of latency and accuracy of the system, it is crucial to establish the robustness of the model against the interference of unrelated regions and its scale invariance. As mentioned in Section 4.1.3, the method employed to diminish the model’s sensitivity to non-vertical regions is the Vertical Denoising population. To further investigate the model’s performance, supplementary event-based recordings have been conducted to verify its efficacy in responding to objects with funny shapes and non-coastline profiles of different sizes.

### 5.4.1 Funny shape dataset

Seven small items (average size  $\sim 1.5$  cm) have been placed on a white desk to demonstrate the model’s robustness to unconventional shapes. The events stream was obtained by recording with a Prophesee HD event-based camera (EVK4 HD [18]) at a height of 3 meters from the objects. Once again, to reduce the size of the input neurons, only the events coming from a central region of  $128 \times 128$  have been collected. The system accurately avoids outputting spikes for the majority of instances (see Figure 8), quantitatively measured at 97.3%. It observes a spike response from 75 CLD maps out of 2760 event-frames input<sup>2</sup>.

### 5.4.2 Augmented dataset

The system has been further assessed by augmenting the Antibes dataset with a harbour and moving boats to resemble a real-world task. The event stream was generated by playing the video on an HD LED monitor (HP P24q G4) and capturing it with an event-based Prophesee HD camera; only the events originating from a central area of  $128 \times 128$  were gathered. Again, the system accurately detects only the coastline, robustly avoiding for the entire duration of the video any of the moving boats or the harbour. Therefore, no spike response, 0 CLD maps out of 550 event-frames. see Figure 8 and the further video showing the VM CLD response at [https://youtu.be/uL13H0\\_BBx8](https://youtu.be/uL13H0_BBx8).

## 6 Discussion

The system demonstrates coastline detection capabilities, achieving a detection rate of 93.48% when utilising a  $\Delta T$  of 50 ms and a threshold of 0.40, thereby generating a CLD map in approximately 50 ms with the possibility of reducing the latency by sacrificing a small amount of accuracy. Unexpectedly, the higher model accuracy, 98.33%, is for a  $\Delta T$  of 30 ms and a threshold of 0.30, which might be an outlier of the system, since in any other case the accuracy increases proportionally with the  $\Delta T$ . While accumulating spikes based on a fixed number may seem advantageous for avoiding the processing of non-movement periods; however, it would result in the complete loss of speed information. As demonstrated in [13], a simple Spiking Elementary Motion Detector (sEMD) can detect motion direction and encode speed through the Mean Firing Rate (MFR) based on consecutive receptive fields. This approach could add complexity to the model in future work. A key advantage of spiking-neural models is their capability for spatiotemporal integration and time-dependent population coding. Therefore, this implementation opted for using a Constant Time Interval (CTI) over a Constant Number of Events (CNE) to ensure accurate spatiotemporal processing across the network, accommodating different time scales within the population. Furthermore, our approach assumes a drone moving at a relatively constant speed (see Section 5). The thresholding

---

<sup>2</sup>Further video showing the VM CLD response at [https://youtu.be/uL13H0\\_BBx8](https://youtu.be/uL13H0_BBx8)

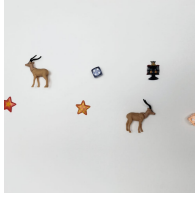
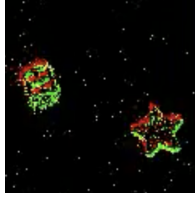
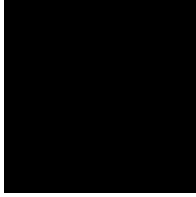

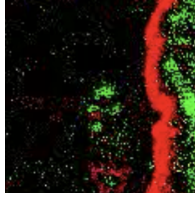
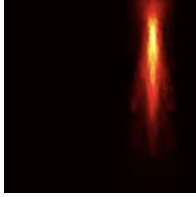
Dataset	Input	ED input	CLD map	CLD maps%
<i>Funny shapes</i>				2.7%
<i>Augmented</i>				0%

Figure 8: Overview of the VM CLD response to funny shapes and non-coastline profiles for the two datasets: *Funny shapes* and *Augmented*. From the left to the right: Dataset name, RGB frame example, event-based input to the model, coastline detection map and the percentage of incorrect spike responses from the model (CLD maps/TOT event-frames). The model responds to only coastline profiles and it is insensitive to funny shapes and boats. Further video showing the VM CLD response at [https://youtu.be/uL13H0\\_BBx8](https://youtu.be/uL13H0_BBx8).

system, in conjunction with vertical denoising, ensures a substantial enhancement in accuracy performance, thereby increasing the model’s sensitivity to coastline profiles and resulting in a higher frequency of high Intersection over Union (IoU) scores. The accuracy percentages of IoU scores exhibit a consistent increase with larger  $\Delta T$  intervals and a decrease in threshold values. The sparse distribution of events enhances detection probabilities by extending the window period of data collection. Lowering the threshold yields higher IoU percentages compared to the average, particularly demonstrating improved performance within the threshold range of 0.3 to 0.5. The lowest threshold exhibits the highest accuracy because it increases the likelihood of predictions exceeding the threshold value. In cases where we obtain low IoU scores, despite this, the model demonstrates accurate detection predictions across all window periods. The system also demonstrates effectiveness without the need for complementary opposite orientations of the VM filter layers for the concave-convex curvature. This is significant and avoids the necessity of increasing the number of layers and neurons for computation. Finally, as expected, we reduced the computational loads and latency from [14] ( $\sim 16$  ms) by utilising only 3 computational layers, resulting in a total of 3 ms (with each layer accounting for 1 ms of the internal SpiNNaker digital clock). Thus, potentially detecting waves with the 92.93% with the lower threshold and  $\Delta T=20$ ms with the possibility of further decreasing the window period reducing the accuracy. The system also demonstrates significant capability in robustness, particularly in its ability to handle irregular shapes and profiles that do not conform to typical coastlines. To further discuss the latency measurements and throughput, given the SpiNNaker system’s minimum time step of 1 ms, input data from the Dynamic Vision Sensor (DVS), which operates in microseconds, must be aggregated over 1 ms before processing. This aggregation causes multiple events within that interval to share the same timestamp, resulting in numerous input and output spikes. A Constant Time Interval (CTI) pseudo-frame method is used to manage excess spikes from highly active regions. This method ensures only one output spike per pixel during the specified accumulation period, reducing potential spikes to one pixel per filter coordinate despite up to 1000 spikes generated over 1 ms. Although the SpiNNaker system may drop packets with high input spike rates, the spike accumulation and pseudo-frame method ensure consistent system performance also in high event rates [58, 46]. Additionally, our approach presumes that the drone maintains a relatively constant speed of 0.28 m/s.

Across the realm of digital neuromorphic platforms such as Loihi, Dynap CNN, and Akida, SpiNNaker stands out for its scalability and flexibility in implementing neuron models and connections among neurons and layers. These neuromorphic chips exhibiting mWs power consumption excel in low-latency processing and efficient power usage for real-time sensory processing. Although SpiNNaker’s power consumption is not the lowest, its performance should be assessed within the context of the specific application.



The total energy consumed by the entire pipeline includes both the energy used to operate the drone and its peripherals, as well as the energy consumed by the SpiNNaker board. Since reducing the energy consumption for driving the drone is outside the scope of this work, the focus is on the energy consumed by the SpiNNaker board. The total energy consumed by the SpiNNaker board can be expressed by the equation in Equation 5

$$P_T = P_I + P_B + (P_N \times n) + (P_S \times s) \quad (5)$$

where  $P_I$  is the power consumed by the SpiNNaker board in idle mode after the boot process, with no applications loaded.  $P_B$  is the baseline power, which includes the power used by SARK and the SpiNNaker API when no neurons are running.  $P_N$  is the power required to simulate a Leaky Integrate-and-Fire (LIF) neuron with a 1 ms time-step.  $n$  is the total number of neurons,  $P_S$  is the energy consumed per synaptic event (when a spike hits a synapse), and  $s$  is the total number of synaptic events (spikes) [54, 53].

Given the sparsity of DVS event data and SpiNNaker’s dynamic energy consumption (8 nJ per synaptic event [54, 53]), it proves more efficient than CPU or GPU processing of frame-based images. SpiNNaker remains a highly flexible and scalable platform that utilises ARM M4 cores, ensuring low latency in multi-board processing configurations. In contrast, transitioning to a multi-board setup on a GPU would significantly slow performance due to limited communication channels between memories.

Table 5 presents the number of spikes emitted by populations during each  $\Delta T = 20$  ms (minimum latency). Except for the input layer coming from the DVS camera, each row corresponds to the spikes generated by all the nine populations at each level of the VM CLD (Figure 2). As illustrated in Table 5, there is a noticeable reduction in the number of spikes as we move from one layer to the next. This reduction can be explained by several factors. The number of neurons in each layer plays a significant role, and layers with fewer neurons naturally produce fewer spikes (see Table 1). The depth of the network affects spike propagation; therefore, deeper layers tend to have stricter spike generation. Finally, the thresholds of the neurons in each population also influence spike activity (Table 2). Neurons in deeper layers or those with higher thresholds require more input to generate spikes, leading to a decrease in overall spike counts. The observed reduction in spike numbers from one layer to the next can be attributed to the interplay between neuron count, network depth, and neuron thresholds, all of which are critical factors in understanding the dynamics of spike propagation in the simulated neural network. Given the average number of spikes recorded during the simulation for a  $\Delta T = 20$  ms, regarding the SpiNNaker dynamic energy consumption rate of 8 nJ per synaptic event [58, 46], the dynamic energy consumption amounts to 7,512 nJ (the average power consumption over 20 ms is approximately 0.3756 mW). This estimation provides valuable insight into the energy efficiency of the model. This work shows the feasibility of employing a neuromorphic device for nearshore and shoreline coastline detection. This work is the first attempt towards the development of more efficient neuromorphic systems in future research and development.

Table 5: Average number of spikes (rounded up) and the consumed energy with a  $\Delta T$  of 20 ms for coastline detection with a drone moving at a constant speed of 0.28 m/s.

Populations	Number of Spikes	Energy Consumption(nJ)
<b>Input</b>	762	6,096
<b>Filter Segments Neurons</b>	151	1,208
<b>Filter Neurons</b>	21	168
<b>Vertical Filter Neurons</b>	5	40
<b>Total</b>	<b>939</b>	<b>7,512</b>

## 7 Conclusion

This implementation is the first neuromorphic attempt at coastline detection taking inspiration from the curved filter used in [14]. The VM kernel filter allowed accurate detection of coastline shapes potentially in about  $\sim 20$  ms running on the neuromorphic SpiNNaker board with 92.93 % accuracy. The shoreline video, collected through the APROVIS3D project, generated frames at 20 FPS, with each frame captured over a 50 ms period. The presented VMCD system demonstrates the ability to achieve coastline detection in as little as 20 ms and up to 50 ms, thereby detecting the coastline even before a single RGB frame is fully captured. However, reducing  $\Delta T$  decreases accuracy across thresholds due to the generation of sparse event frames. Given the application context of nearshore monitoring and coastline detection, where the drone operates at a moderate speed, this model presents a valuable alternative to frame-based methods. It allows for coastline detection within 50 ms, whereas a classical frame-based camera would only produce a frame within that time frame, without even starting the processing. Additionally, the model consumes significantly less power, as little as 0.3756 mW, with a latency of 20 ms. As expected, a future investigation of this work would include the evaluation with a more complete real-world dataset at different heights and landscapes. Further improvement could involve incorporating event polarity as a crucial asset for determining wave direction, thereby facilitating the detection of visual field regions associated with the coast and those corresponding to the sea. One additional idea for future work regarding the model is its limited capability to handle rotational fields of view. A potential solution could involve reorienting the drone using a coastline orientation detector, thereby expanding the orientations of the VM layers. However, this may increase the number of neurons required, possibly exceeding the capacity of a single SpiNNaker board. Furthermore, the method is designed to enable control of the drone to maintain alignment with the vertical line of the coastline. Therefore, another potential solution would be to rotate the drone until it achieves a high score in coastline detection. The authors' aspiration is that this model will pave the way for more sophisticated shoreline detection implementations on neuromorphic hardware. Additionally, the model's robustness to a wide variety of shapes demonstrates its potential in diverse applications, such as road and river monitoring. It proves to serve as the foundation for innovative low-latency and low-power monitoring technologies.

## 8 Acknowledgement

We thank Jean Martinet, the coordinator of the Aprovis3D project and Professor at the Université Côte d'Azur, for providing us with the RGB recordings of the coastline monitoring, which enabled us to generate and finalize this work.

This project has received the support of IRCICA (Univ. Lille, CNRS, USR 3380 - IRCICA - Institut de Recherche sur les Composants logiciels et matériels pour l'Information et la Communication Avancée, F-59000 Lille, France) and the Luxant-ANVI industrial chair (Métropole Européenne de Lille and Univ. Lille).

## Author contributions statement

M.F. has been initially selected to work on the APROVIS3D which has conceived the task application and entirely supports this work. G.D. conceived the main idea behind the research novelty of the work. G.D. and M.F. developed the theory for the VM CLD model implementation. M.F. developed the code and the entire pipeline on the neuromorphic platform, with help from G.D.; M.F. and G.D. designed the experiments with supervision from P.B.; M.F. conducted experiments. M.F. and G.D. analysed the experimental results. M.F. and G.D. wrote the manuscript. P.B. gave valuable feedback and helped edit the manuscript. G.D. made the supplementary video accompanying this work.

## References

- [1] Rafael Almar et al. "Wave runup video motion detection using the Radon Transform". In: *Coastal Engineering* 130 (2017), pp. 46–51.

- [2] *Aprovis3D*. en. URL: <http://hassium.geo.uniwa.gr/> (visited on 01/17/2024).
- [3] Maria Christina Barbosa de Araújo and Monica Ferreira da Costa. “Visual diagnosis of solid waste contamination of a tourist beach: Pernambuco, Brazil”. In: *Waste Management* 27.6 (2007), pp. 833–839.
- [4] Eduard Ariza, José A Jiménez, and Rafael Sardá. “Seasonal evolution of beach waste and litter during the bathing season on the Catalan coast”. In: *Waste management* 28.12 (2008), pp. 2604–2613.
- [5] F Asensio-Montesinos, G Anfuso, and AT Williams. “Beach litter distribution along the western Mediterranean coast of Spain”. In: *Marine pollution bulletin* 141 (2019), pp. 119–126.
- [6] Sotirios N Aspragkathos, George C Karras, and Kostas J Kyriakopoulos. “A Hybrid Model and Data-Driven Vision-Based Framework for the Detection, Tracking and Surveillance of Dynamic Coastlines Using a Multirotor UAV”. In: *Drones* 6.6 (2022), p. 146.
- [7] Sotirios N Aspragkathos, George C Karras, and Kostas J Kyriakopoulos. “A visual servoing strategy for coastline tracking using an unmanned aerial vehicle”. In: *2022 30th Mediterranean Conference on Control and Automation (MED)*. IEEE. 2022, pp. 375–381.
- [8] Sotirios N Aspragkathos et al. “An Event-Based Tracking Control Framework for Multirotor Aerial Vehicles Using a Dynamic Vision Sensor and Neuromorphic Hardware”. In: *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2023, pp. 6349–6355.
- [9] Arindam Basu et al. “Spiking Neural Network Integrated Circuits: A Review of Trends and Future Directions”. In: *2022 IEEE Custom Integrated Circuits Conference (CICC)*. 2022, pp. 1–8. DOI: 10.1109/CICC53496.2022.9772783.
- [10] Taha Benarbia and Kyandoghere Kyamakya. “A literature review of drone-based package delivery logistics systems and their implementation feasibility”. In: *Sustainability* 14.1 (2021), p. 360.
- [11] Ed Bormashenko, A Musin, and R Grynyov. “Scaling law governing the roughness of the swash edge line”. In: *Scientific Reports* 4.1 (2014), p. 6243.
- [12] Adam M. Collins et al. “Automated Extraction of a Depth-Defined Wave Runup Time Series From Lidar Data Using Deep Learning”. In: *IEEE Transactions on Geoscience and Remote Sensing* 61 (2023), pp. 1–13. DOI: 10.1109/TGRS.2023.3244488.
- [13] Giulia D’Angelo et al. “Event-based eccentric motion detection exploiting time difference encoding”. In: *Frontiers in neuroscience* 14 (2020), p. 451.
- [14] Giulia D’Angelo et al. “Event driven bio-inspired attentive system for the iCub humanoid robot on SpiN-Naker”. In: *Neuromorphic Computing and Engineering* 2.2 (2022), p. 024008.
- [15] Mike Davies et al. “Loihi: A neuromorphic manycore processor with on-chip learning”. In: *Ieee Micro* 38.1 (2018), pp. 82–99.
- [16] Andrew Davison et al. “PyNN: a common interface for neuronal network simulators”. In: *Frontiers in Neuroinformatics* 2 (2009). ISSN: 1662-5196. DOI: 10.3389/neuro.11.011.2008.
- [17] Silvio Del Pizzo et al. “Assessment of shoreline detection using UAV”. In: *Proceedings of the 2020 IMEKO TC-19 International Workshop on Metrology for the Sea (MetroSea 2020), Naples, Italy*. 2020, pp. 5–7.
- [18] *Event Camera Evaluation Kit 4 HD IMX636 Prophesee-Sony*. en-US. URL: <https://www.prophesee.ai/event-camera-evk4/> (visited on 01/16/2024).
- [19] Steve Furber and Petruț Bogdan. *Spinnaker—a spiking neural network architecture*. Now publishers, 2020.
- [20] Steve B Furber et al. “The spinnaker project”. In: *Proceedings of the IEEE* 102.5 (2014), pp. 652–665.
- [21] Steve B. Furber et al. “Overview of the SpiNNaker System Architecture”. In: *IEEE Transactions on Computers* 62.12 (2013), pp. 2454–2467. DOI: 10.1109/TC.2012.142.
- [22] Guillermo Gallego et al. “Event-based vision: A survey”. In: *IEEE transactions on pattern analysis and machine intelligence* 44.1 (2020), pp. 154–180.

- [23] Suman Ghosh et al. “Event-driven proto-object based saliency in 3D space to attract a robot’s attention”. In: *Scientific reports* 12.1 (2022), p. 7645.
- [24] Arren Glover and Chiara Bartolozzi. “Robust visual tracking with a freely-moving event camera”. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2017, pp. 3769–3776.
- [25] Zihui Guo et al. “Motion saliency based hierarchical attention network for action recognition”. In: *Multimedia Tools and Applications* 82.3 (2023), pp. 4533–4550.
- [26] Matthew Harry et al. “Observation of surf zone wave transformation using LiDAR”. In: *Applied Ocean Research* 78 (2018), pp. 88–98.
- [27] Sheng Hui et al. “Coastline extraction based on multi-scale segmentation and multi-level inheritance classification”. In: *Frontiers in Marine Science* 9 (2022), p. 1031417.
- [28] Massimiliano Iacono et al. “Proto-object based saliency for event-driven cameras”. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2019, pp. 805–812.
- [29] Lauren Nicole Kim et al. “Observations of beach change and runup, and the performance of empirical runup parameterizations during large storm events”. In: *Coastal Engineering* (2023), p. 104357.
- [30] Paul Kirkland et al. “UAV detection: a STDP trained deep convolutional spiking neural network retinaneuromorphic approach”. In: *Artificial Neural Networks and Machine Learning–ICANN 2019: Theoretical Neural Computation: 28th International Conference on Artificial Neural Networks, Munich, Germany, September 17–19, 2019, Proceedings, Part I 28*. Springer. 2019, pp. 724–736.
- [31] P Santosh Kumar et al. “Brain tumor segmentation of the FLAIR MRI images using novel ResUnet”. In: *Biomedical Signal Processing and Control* 82 (2023), p. 104586.
- [32] Chuan Li et al. “Observations of extreme wave runup events on the US Pacific Northwest coast”. In: *Natural Hazards and Earth System Sciences* 23.1 (2023), pp. 107–126.
- [33] Alejandro Linares-Barranco et al. “Low latency event-based filtering and feature extraction for dynamic vision sensors in real-time FPGA applications”. In: *IEEE Access* 7 (2019), pp. 134926–134942.
- [34] Marco Luppichini et al. “A new beach topography-based method for shoreline identification”. In: *Water* 12.11 (2020), p. 3110.
- [35] Kévin Martins et al. “High-resolution monitoring of wave transformation in the surf zone using a LiDAR scanner array”. In: *Coastal Engineering* 128 (2017), pp. 37–43.
- [36] Balmukund Mishra et al. “Drone-surveillance for search and rescue in natural disaster”. In: *Computer Communications* 156 (2020), pp. 1–10.
- [37] Marco Monforte et al. “Exploiting Event Cameras for Spatio-Temporal Prediction of Fast-Changing Trajectories”. In: *2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*. IEEE. 2020, pp. 108–112.
- [38] João Gama Monteiro et al. “Novel approach to enhance coastal habitat and biotope mapping with drone aerial imagery analysis”. In: *Scientific Reports* 11.1 (2021), p. 574.
- [39] Eustace Painkras et al. “SpiNNaker: A 1-W 18-Core System-on-Chip for Massively-Parallel Neural Network Simulation”. In: *IEEE Journal of Solid-State Circuits* 48.8 (2013), pp. 1943–1953. DOI: 10.1109/JSSC.2013.2259038.
- [40] Federico Paredes-Vallés et al. “Fully neuromorphic vision and control for autonomous drone flight”. In: *arXiv preprint arXiv:2303.08778* (2023).
- [41] LA Plana. *AppNote 8 - Interfacing AER devices to SpiNNaker using an FPGA*. URL: <http://spinnakermanchester.github.io/docs/spinn-app-8.pdf>.
- [42] Carly E Portch et al. “Wave runup and inundation dynamics on a perched beach”. In: *Geomorphology* 435 (2023), p. 108751.

- [43] Christoph Posch, Daniel Matolin, and Rainer Wohlgenannt. “A QVGA 143 dB dynamic range frame-free PWM image sensor with lossless pixel-level video compression and time-domain CDS”. In: *IEEE Journal of Solid-State Circuits*. Vol. 46. 2011, pp. 259–275. ISBN: 9781424460342. DOI: 10.1109/JSSC.2010.2085952. URL: <http://ieeexplore.ieee.org/document/5648367/>.
- [44] Matteo Postacchini, Lorenzo Melito, and Giovanni Ludeno. “Nearshore observations and modeling: Synergy for coastal flooding prediction”. In: *Journal of Marine Science and Engineering* 11.8 (2023), p. 1504.
- [45] Oliver Rhodes et al. “sPyNNaker: A Software Package for Running PyNN Simulations on SpiNNaker”. In: *Frontiers in Neuroscience* 12 (2018). ISSN: 1662-453X. DOI: 10.3389/fnins.2018.00816.
- [46] Juan Pablo Romero Bermudez et al. “A high-throughput low-latency interface board for SpiNNaker-in-the-loop real-time systems”. In: *Proceedings of the 2023 International Conference on Neuromorphic Systems*. 2023, pp. 1–8.
- [47] Alexander F Russell et al. “A model of proto-object based saliency”. In: *Vision research* 94 (2014), pp. 1–15.
- [48] Yannick Schneider et al. “Neuromorphic Optical Flow and Real-time Implementation with Event Cameras”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 4128–4137.
- [49] Oscar Bowen Schofield, Nicolai Iversen, and Emad Ebeid. “Autonomous power line detection and tracking system using UAVs”. In: *Microprocessors and Microsystems* 94 (2022), p. 104609.
- [50] Teresa Serrano-Gotarredona and Bernabé Linares-Barranco. “A  $128 \times 128$  1.5% Contrast Sensitivity 0.9% FPN  $3 \mu\text{s}$  Latency 4 mW Asynchronous Frame-Free Dynamic Vision Sensor Using Transimpedance Preamplifiers”. In: *IEEE Journal of Solid-State Circuits* 48.3 (2013), pp. 827–838.
- [51] Ghada A Shadeed, Mohammed A Tawfeeq, and Sawsan M Mahmoud. “Automatic medical images segmentation based on deep learning networks”. In: *IOP Conference Series: Materials Science and Engineering*. Vol. 870. 1. IOP Publishing. 2020, p. 012117.
- [52] Zhenfeng Shao et al. “Saliency-aware convolution neural network for ship detection in surveillance video”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 30.3 (2019), pp. 781–794.
- [53] Evangelos Stamatias et al. “Power analysis of large-scale, real-time neural networks on SpiNNaker”. In: *The 2013 international joint conference on neural networks (IJCNN)*. IEEE. 2013, pp. 1–8.
- [54] Evangelos Stamatias et al. “Scalable energy-efficient, low-latency implementations of trained spiking deep belief networks on spinnaker”. In: *2015 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2015, pp. 1–8.
- [55] Indar Sugiarto et al. “High performance computing on spinnaker neuromorphic platform: A case study for energy efficient image processing”. In: *2016 IEEE 35th International Performance Computing and Communications Conference (IPCCC)*. IEEE. 2016, pp. 1–8.
- [56] E Tamassoki, H Amiri, and Z Soleymani. “Monitoring of shoreline changes using remote sensing (case study: coastal city of Bandar Abbas)”. In: *IOP conference series: earth and environmental science*. Vol. 20. 1. IOP publishing. 2014, p. 012023.
- [57] Nico Valentini et al. “New algorithms for shoreline monitoring from coastal video systems”. In: *Earth Science Informatics* 10 (2017), pp. 495–506.
- [58] Sacha J Van Albada et al. “Performance comparison of the digital neuromorphic hardware SpiNNaker and the neural network simulation software NEST for a full-scale cortical microcircuit model”. In: *Frontiers in neuroscience* 12 (2018), p. 291.
- [59] Antonio Vitale et al. “Event-driven vision and control for UAVs on a neuromorphic chip”. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 103–109.
- [60] Paul Webb. *Introduction to oceanography*. Roger Williams University, 2021.
- [61] Wojciech Wilkowski et al. “The use of unmanned aerial vehicles (drones) to determine the shoreline of natural watercourses”. In: *Journal of Water and Land Development* 35.1 (2017), p. 259.

- [62] Jiayu Xing et al. “Autonomous Power Line Inspection with Drones via Perception-Aware MPC”. In: *arXiv preprint arXiv:2304.00959* (2023).
- [63] Song Zhang et al. “Learning to see in the dark with events”. In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*. Springer. 2020, pp. 666–682.
- [64] Hong Zhou, Howard S Friedman, and Rüdiger von der Heydt. “Coding of Border Ownership in Monkey Visual Cortex”. In: *The Journal of Neuroscience* 20.17 (2000), pp. 6594–6611.
- [65] Xixuan Zhou et al. “An Overview of Coastline Extraction from Remote Sensing Data”. In: *Remote Sensing* 15.19 (2023), p. 4865.
- [66] Ming Zong et al. “Spatial and temporal saliency based four-stream network with multi-task learning for action recognition”. In: *Applied Soft Computing* 132 (2023), p. 109884.