# - APROVIS3D -

Analog PROcessing of bioinspired VIsion Sensors for 3D reconstruction

| | | |
|---|---|---|
| Document Reference: | | |
| Title:<br>**D3.3 Report on coast-line detection using the foveated sensor (INT)** | | |
| Contractor:<br>**INT** | | |
| Prepared by:<br>**Mazdak Fatahi, Antoine Grimaldi and Laurent Perrinet** | | |
| Document Type: **Deliverable** | | |
| Version: **1.1** | | Pages: **13** |
| Classification: **External document** | | |

## Document Track

| Version | Date | Remarks and Authors |
|---|---|---|
| 1.0 | 12/12/2023 | First Draft (M. Fatahi, A. Grimaldi & L. Perrinet – INT) |
| 1.1 | 20/12/2023 | Last version (J. Martinet – UCA) |
| | | |

## Authors

| | Role / Function | Name | Organisation |
|---|---|---|---|
| Prepared by | WP3 Contributors | A. Grimaldi & L. Perrinet | INT |
| Checked by | Quality | J. Martinet | UCA |
| Released by | WP3 Leader | L. Perrinet | INT |
| Approved by | Project Coordinator | J. Martinet | UCA |

# TABLE OF CONTENTS

# FIGURES

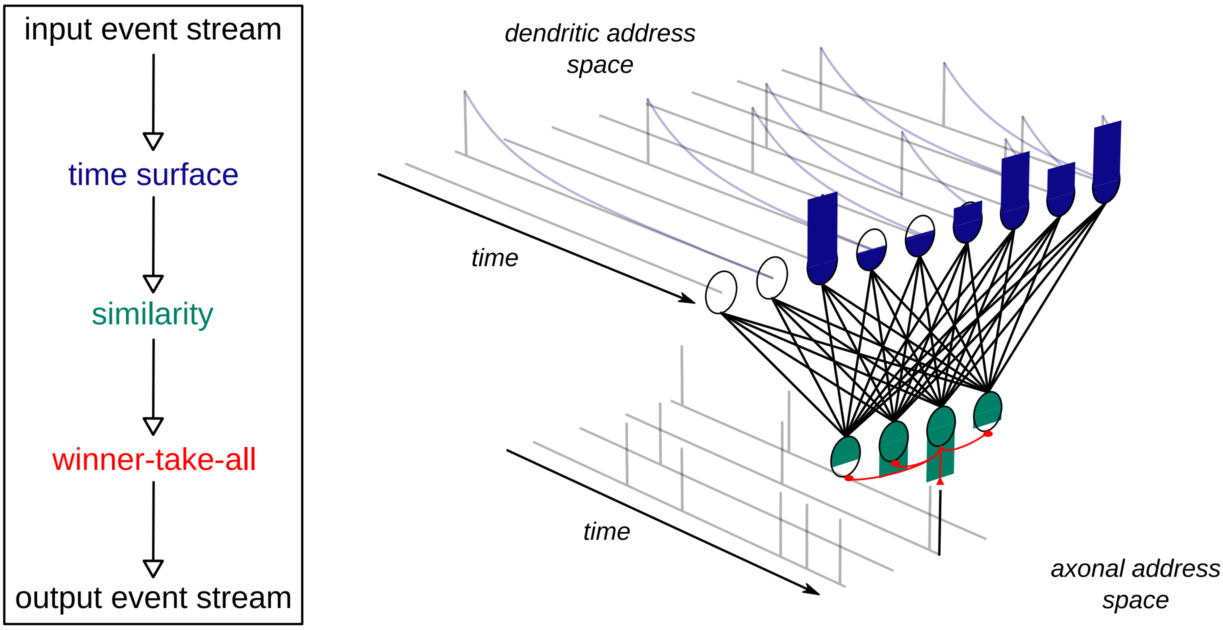



*Figure 1: **Illustration of the core computation made within one layer of the HOTS algorithm.** On the top of the plot, we show the dendritic stream of events convolved by an exponential decay which forms the time surface. Time surfaces are computed at the timestamp of each event/spike. The time surface at present is represented with the colored bar plot on the top. In the vertical slice, computations made within one layer at time $t_i$ are illustrated. The time surface is compared to all the kernels of the layer with the similarity measure resulting in the membrane potential of the postsynaptic neuron represented in green. As an illustration, the layer contains only 4 neurons associated to 4 different kernels and with 10 dendritic inputs. At last, a winner-take-all rule (or argmax non-linearity) will choose at time $t_i$ the most activated neuron. This will emit a spike and prevent the others from being activated through lateral inhibitions (in red). Note that for each event as input of the layer, a new event will be emitted with the same timing as the incoming event. Figure adapted from [Grimaldi et al (2023)](#).*

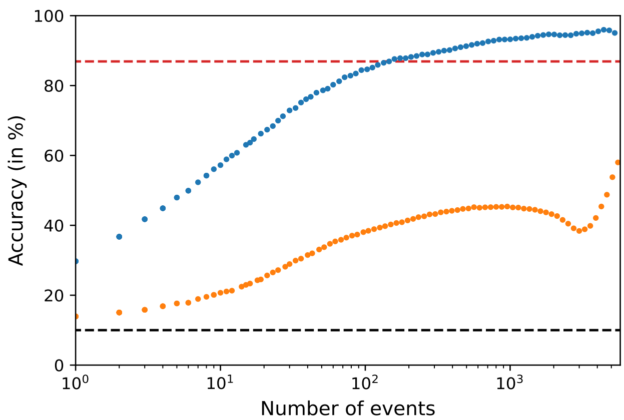



*Figure 2: **Always-on classification performance of the SNN solution**. Our SNN solution allows to perform a classification at any time and we show here the average accuracy computed on the N-MNIST dataset with respect to the number of events since the beginning of the stream. The red horizontal dashed line shows the classification performance for the offline method described in [Lagorce et al. 2017.](#) In blue, we show the results for our online classification algorithm and in orange, as a control, a MLR applied on the raw event stream (without passing through the HOTS network). This shows a rapid classification after a few hundreds events.*

# 1. Disclaimer

This deliverables describes activities in the project dedicated to the coastline detection, however with a standard non-foveated sensor because of the delayed availability of the foveated sensor. Therefore, all content described here is based on a sensor with constant resolution.

## 2. Introduction

We developed a SNN-based machine learning adapted for DVS signal and report here our progress in applying it to coastline detection. To achieve this, we first extended an existing event-based algorithm [Lagorce et al., 2017], which introduced novel spatio-temporal features as a Hierarchy Of Time-Surfaces (HOTS). Built from asynchronous events acquired by a neuromorphic camera, these time surfaces allow to code the local dynamics of a visual scene and to create an efficient event-based pattern recognition architecture. Our first contribution was to add a homeostatic gain control on the activity of neurons to improve the learning of spatio-temporal patterns. A second contribution is to draw an analogy between the HOTS algorithm and Spiking Neural Networks (SNN). Following that analogy, our last contribution is to modify the classification layer and remodel the offline pattern categorization method previously used into an online and event-driven one. This classifier uses the spiking output of the network to define novel time surfaces and we then perform online classification with a neuromimetic implementation of a multinomial logistic regression. Not only do these improvements increase consistently the performances of the network, they also make this event-driven pattern recognition algorithm online and bio-realistic. This work is currently reviewed by the journal *Neural Networks* and is available as a preprint. This algorithm was tested on different datasets for symbol recognition and aim at being applied to coastline detection. This classification for each event can easily be adapted to a segmentation algorithm allowing to distinguish between the sea and the ground in order to locate the coastline. This report presents some results on the datasets developed during the APROVIS3D project.

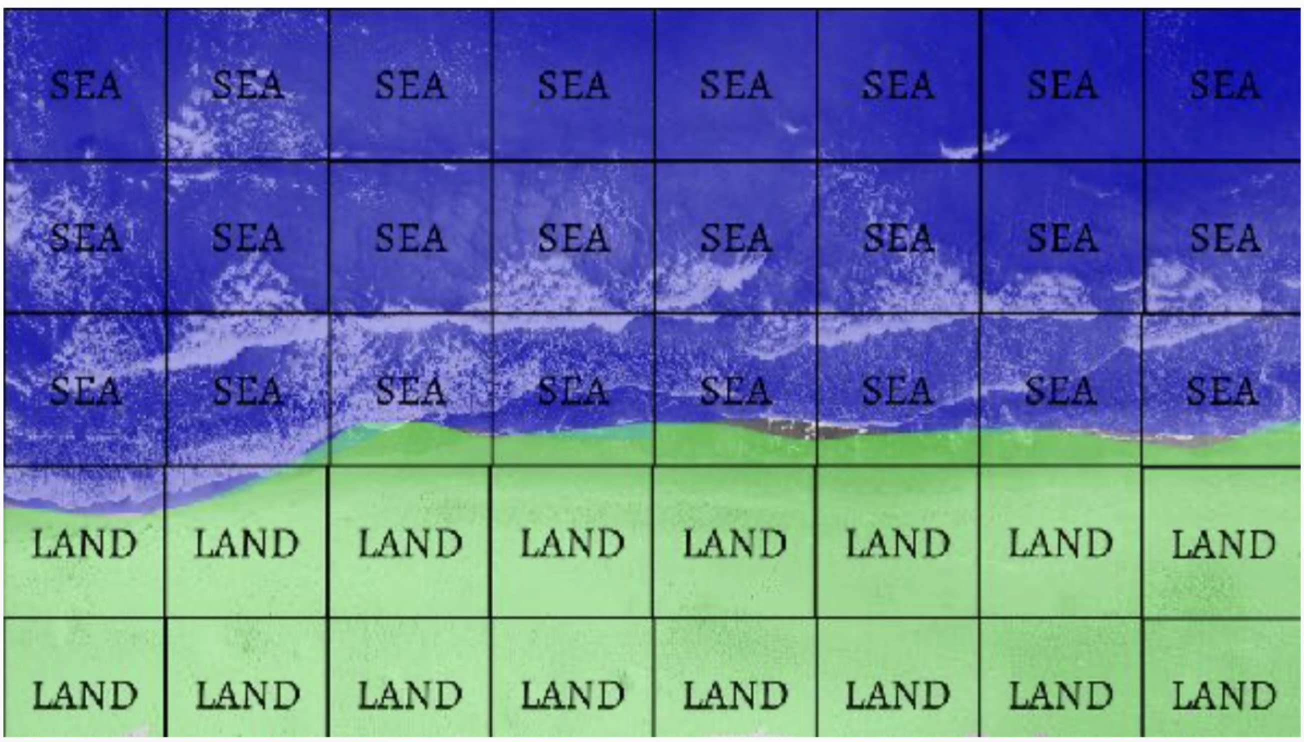



*Figure 3: From detection to segmentation: In this WP, we used patch-based detection to determine which areas of the visual field are part of the ground or land and which are part of the sea. The interface between the two categories gives the outline of the coastline.*

# 3. Documentation

## 3.1 Applicable and Referenced Documents

| # | Id | Description | Identifier (Ed Rev) | Date |
|---|---|---|---|---|
| AD1 | FPP | Full Project Proposal | 1.0 | 15.01.2019 |
| | | | | |

## 3.2 Glossary and Terminology

| Acronym | Definition |
|---|---|
| WP | Work Package |
| SNN | Spiking Neural Network |
| ML | Machine Learning |
| HOTS | Hierarchy Of Time-Surfaces |
| MLR | Multinomial Logistic Regression |
| | |

# 4. Contents

## 4.1 Methodology: An already existing event-based algorithm

Our method is based on an existing algorithm for event-based pattern recognition: Lagorce et al. 2017. In this work, the authors define time surfaces, a spatio-temporal object useful to treat the dynamics within the stream of events. With a hierarchical network able to learn prototypical time surfaces, the stream of events is transformed and used as an accumulation of spatio-temporal features to perform object classification. Its name stems from this Hierarchy Of Time Surfaces (HOTS). Major advantages are the event-triggered, online and unsupervised training of the hierarchical network and the transformation of the flux of events into a more complex flux of events allowing stacking of layers. This method is easily transferable to neuromorphic hardware considering the event-based nature of the computations.

## 4.2 Improvements

We identified two major drawbacks for this interesting algorithm: its learning of time surfaces is very sensitive to initialization and the classification is done offline, once the stream of events is treated by the network, by assessing the activation of the last layer through histogram comparison.
To deal with the variability in the clustering phase, we introduce a homeostatic gain control to regulate the activity of the neurons within one layer. Regarding the past activity of the neurons, one is prevented or encouraged to spike wether if it did or did not spike enough compared to neighbouring neurons. Results on the improvements linked to homeostasis are reported in a conference article: Grimaldi et al. 2021.
Then, we introduce a new classification layer that allows event-per-event online classification. The spiking mechanism of this classification layer is similar to one layer of the HOTS network and is illustrated in Figure 1. This always-on classifier is performed by a MLR and results are reported in this preprint: Grimaldi et al. 2023. In this article, we also demonstrate formally the relationship between this event-based, always-on classification algorithm and a SNN. Again, we show that this algorithm is implementable in a neuromorphic device.

## 4.3 Results on the N-MNIST dataset

We first report classification results on a widely used dataset: the N-MNIST dataset, which is a neuromorphic version of the well-known MNIST dataset. We show that with only a small number of events in the event stream, the performance of our method outperforms the previous offline method. By the end of the event stream, its performance is competitive with the state of the art. This enables ultra-fast object categorisation, and we plan to apply this method to a segmentation task for coastline detection.

## 4.4 Patch-based segmentation with HOTS

We have developed a bio-inspired machine learning algorithm for event-based vision. This algorithm essentially uses a SNN and can be implemented on a neuromorphic chip. Organised in a hierarchical structure similar to the visual cortex, our network is able to recognise spatio-temporal prototypical patterns through time surfaces. These time surfaces exploit the delays between the last recorded events and provide an interesting method for processing event streams. We aim to apply this method to a segmentation task between 'ground' and 'sea' patches, leading to the detection of the coastline.

1. Synthetic data from a flight simulator

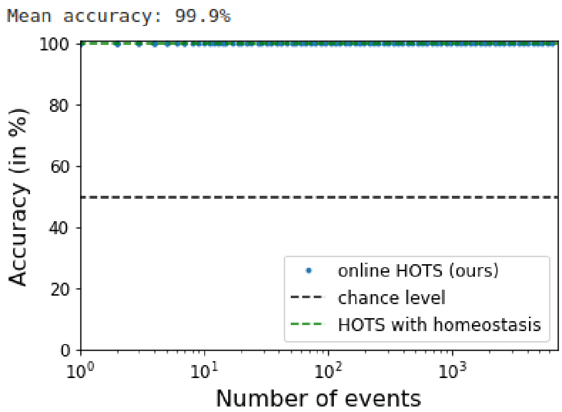
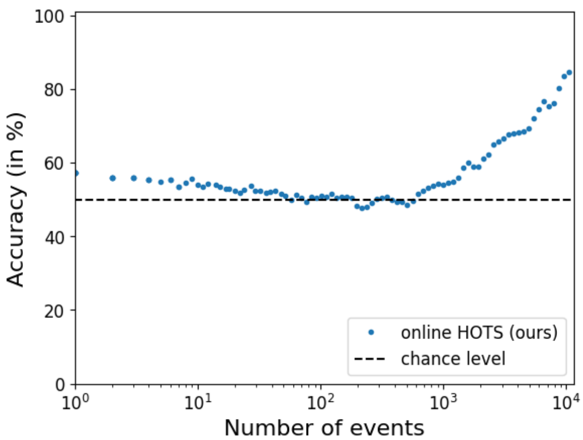


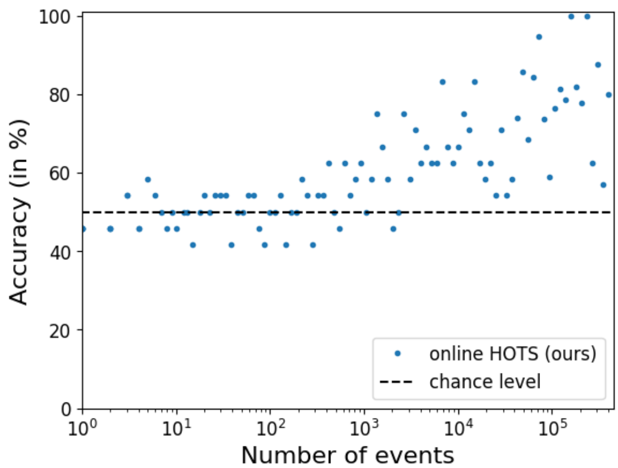

*Figure 4: Accuracy of the SNN on synthetic data.*

We started by applying the classification layer (MLR) of our model to synthetically generated Dynamic Vision Sensor (DVS) data from the NTUA flight simulator. The simulator output was divided into 16x16 pixel patches of 1 second each. Using only the single MLR layer gave a near perfect test accuracy of approximately 99.9%. However, on further analysis of the synthetic data, we found that classification could be achieved simply by exploiting differences in raw event density between the sea and ground textures. This raised concerns that the model might be overfitting to the biases inherent in the simplified synthetic data, rather than learning robust discriminative visual features. To properly assess the real-world applicability of the approach, we recognised the need for a more realistic event-based dataset generator capable of capturing target classes under more challenging and ambiguous conditions that would remove synthetic biases. This would allow us to determine whether the method could generalise beyond the simulated training domain.

2. Event generated from RGB recordings

RGB recordings of sea and ground scenes were transformed into events thanks to an already existing python package (provided by partner UCA). The recordings are divided into 1 second sequences to gather a training set of 69 samples and a testing set of 24 samples. We applied the HOTS algorithm with two layers to the event streams to obtain the transformed event streams that will be used for classification. We show that classification can be performed with (left figure) or without (right figure) the creation of 16x16 patches that goes inline with the multiscale feature of the foveated sensor. We observe an increasing of classification performance when the number of events increases. For the patched based classification we obtain 68.1% accuracy and for the event-based classification 74.1%.

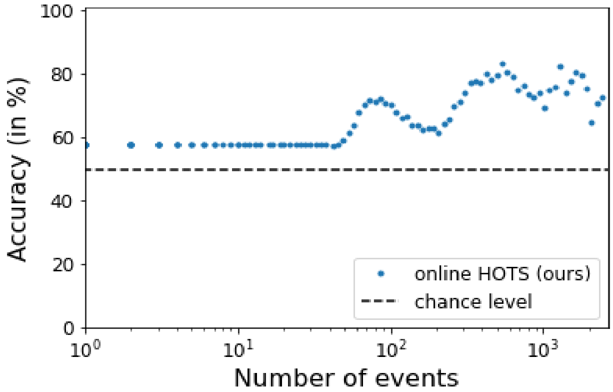


If we put a confidence threshold on the classification decision we highly increase the performances of the method (88.6% correct classification) but that method does not guarantee a decision for each event, rather for each event with a high confidence.

3. Application to a signal obtained from a DVS

We tested this retinotopic mapping method on a dataset recorded by partner NTUA using a Dynamic Vision Sensor (DVS). The NTUA DVS dataset allows classification of visual events as representing either sea or ground. Applying our HOTS spiking neural network to this data achieved 77.9% accuracy using only 100 events. This early classification performance was able to infer coastline locations through a straightforward regression algorithm. However, the current dataset is still relatively small and would not suffice for deployment in a full production system running on our event-driven flying robot platform. Nonetheless, the promising results provide evidence that extending HOTS to classify larger neuromorphic vision datasets could enable effective visual processing applications in real time. While more experimentation is needed at larger scales, this work demonstrates the potential of the spike-based approach.



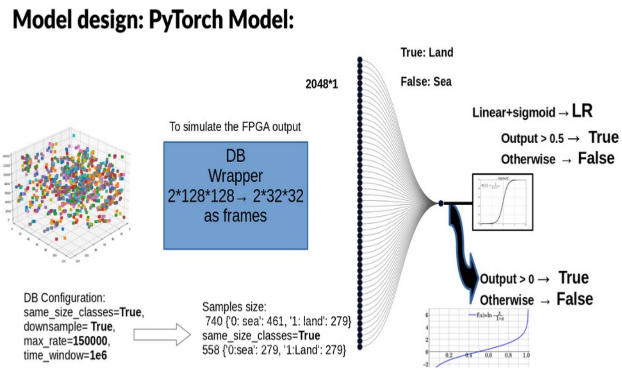

## 4.5 SpiNNaker Implementations

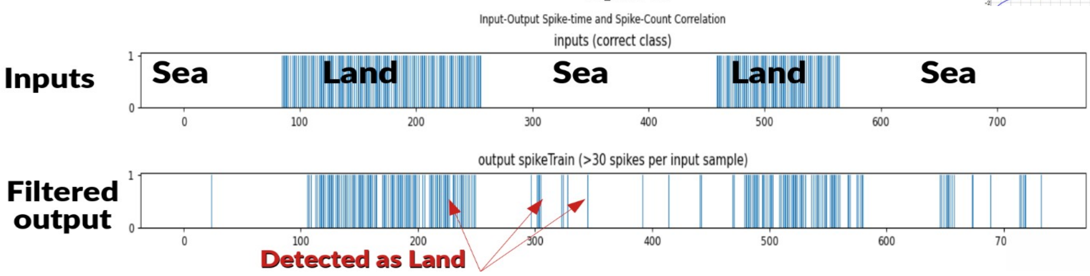
In light of the limitations of SpiNN-3 and the imperative considerations regarding drone control timing constraints, we explored the feasibility of employing smaller models.

Our initial experimentation involved a one-layer classifier, wherein the model was trained on a computer, and subsequently, the weights were transferred to SpiNN-3.

1. Binary Classifier

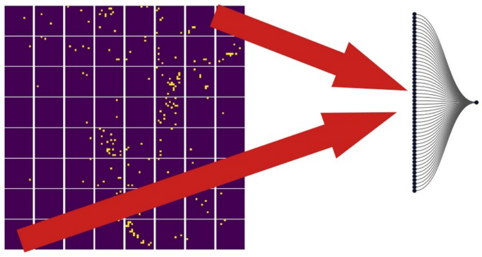
Preliminary findings indicate that when the drone is flying over land, an increased number of spikes are observed. However, this alone proves insufficient for effectively tracking coastlines. Recognizing the need for enhanced resolution, we progressed to a patch-based model.

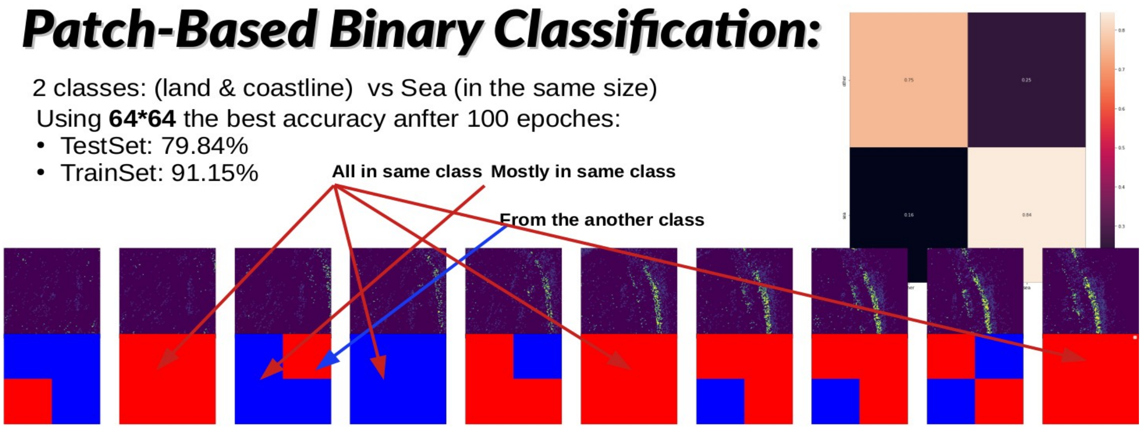


2. Patch-Based Segmentation( without HOTS)

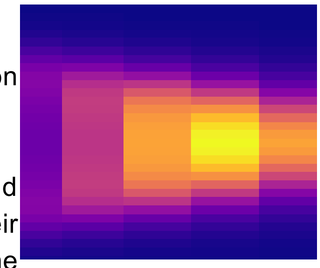
Illustrated in Figure.2, this approach involves spatially dividing each input sample into multiple patches for training a binary classifier. The model's performance is notably influenced by the patch-size parameter. Optimal results were achieved with a size of 64 by 64 (refer to Figure.2). Despite achieving commendable accuracy and recall, the utilization of such a large patch size is not conducive to our specific objectives.





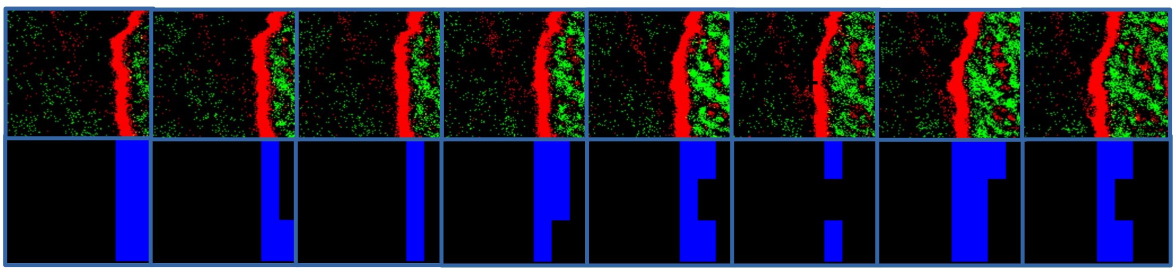

3. Attention-based Model

Considering the substantial number of events generated by moving objects, we presume that the waves forming along the shoreline can serve as reliable indicators for coastline detection. Additionally, since positive events occurring in front of the waves distinctly indicate their direction of movement, we exclusively utilize these positive events to conserve space on the SpiNNaker board.



To focus on the region of interest (the moving waves), we employ a series of Von Mises filters (VM) with varying scales and orientations.

We utilized VM filters with 3 scales (16, 32 and 64) and 3 orientations (-15◦, 0◦ and 15◦) to capture different regions of the moving coastline. The outputs from their corresponding neural populations are integrated in a Grouping Population on the SpiNNaker board. So, establishing appropriate inhibitory connections enables a competition among various scales and orientations and the SpiNNaker board's ultimate output illustrates the trajectory of the coastline.



## 4.6 Perspectives

We have made progress in applying our SNN methodology to different types of vision data, with promising initial results in each case. Initial testing on synthetic DVS output from the NTUA flight simulator achieved near perfect classification accuracy. This validated the approach, but highlighted potential overfitting issues. To address this, we then evaluated performance on real RGB video data converted to events, achieving a small improvement in accuracy. More recently, we presented results using the raw output of a dynamic vision sensor, achieving our highest classification rate to date without any pre-processing.
At the same time, Mazdak Fatahi's work has demonstrated the feasibility of transferring our spiking neural network technology to low-power neuromorphic hardware - specifically the SpiNNaker 3 (SPIN-3) board. Deploying even early versions of our models on this specialised asynchronous substrate confirms the potential for low-latency embedded implementation.
Looking ahead, we aim to continue optimising our SNN and event-based datasets in parallel. Larger, more diverse datasets will certainly improve real-world generalisation. At the same time, further algorithmic and architectural refinement can help exploit the capabilities of neuromorphic hardware. We expect these two research directions - data and architecture - to be mutually beneficial as development continues in collaboration between our group and collaborators such as Mazdak Fatahi. The combination of increasingly powerful SNNs with specialised neuromorphic platforms holds great promise for the realisation of truly event-driven machine perception.

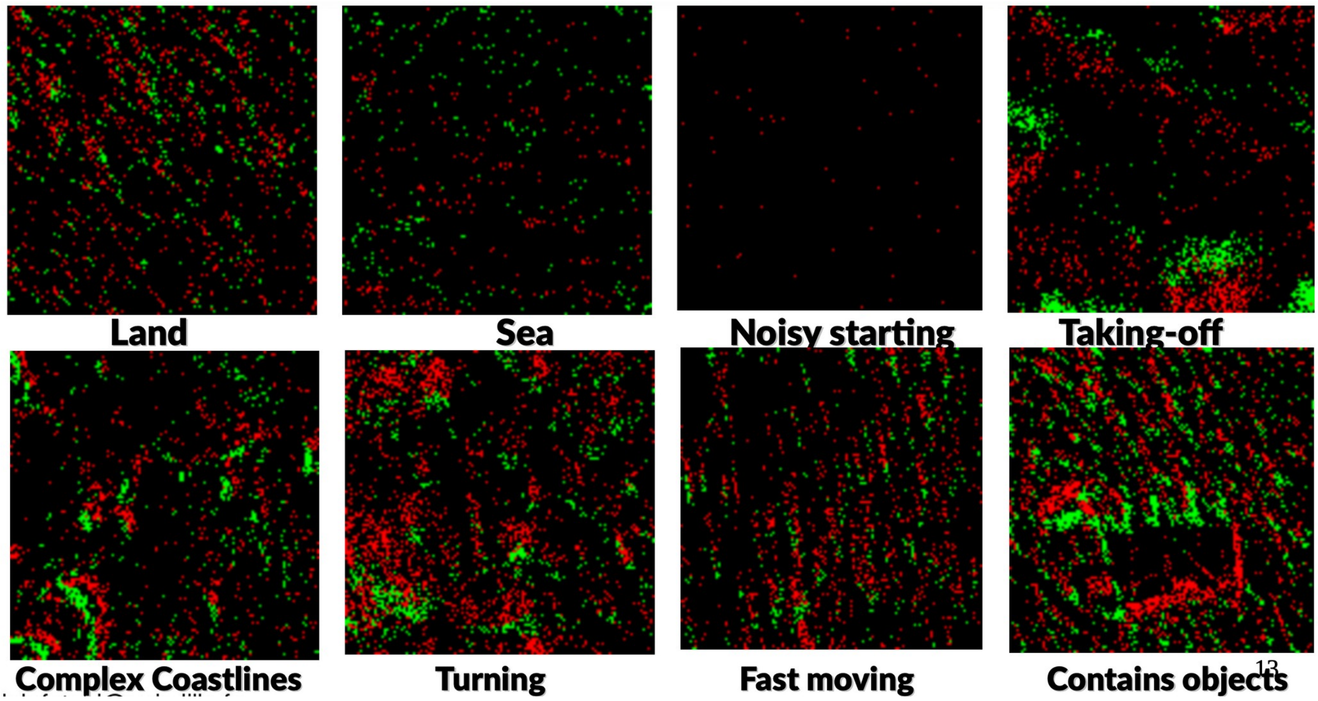



*Figure 5: Transfer of the spiking neural network methods to the SPINN-3 board by Mazdak Fatahi.*