SpikeSEG: Spiking Segmentation via STDP Saliency Mapping

Paul Kirkland
Neuromorphic Sensing and Processing Lab
University of Strathclyde
Glasgow, Scotland, UK
paul.kirkland@starth.ac.uk

Gaetano Di Caterina
Neuromorphic Sensing and Processing Lab
University of Strathclyde
Glasgow, Scotland, UK
gaetano.di-caterina@strath.ac.uk

John Soraghan
Neuromorphic Sensing and Processing Lab
University of Strathclyde
Glasgow, Scotland, UK
j.soraghan@strath.ac.uk

George Matich
Leonardo
London, UK
george.matich
@leonardocompany.com

***Abstract*—Taking inspiration from the structure and behaviour of the human visual system and using the Transposed Convolution and Saliency Mapping methods of Convolutional Neural Networks (CNN), a spiking event-based image segmentation algorithm, SpikeSEG is proposed. The approach makes use of both spike-based imaging and spike-based processing, where the images are either standard images converted to spiking images or they are generated directly from a neuromorphic event driven sensor, and then processed using a spiking fully convolutional neural network. The spiking segmentation method uses the spike activations through time within the network to trace back any outputs from saliency maps, to the exact pixel location. This not only gives exact pixel locations for spiking segmentation, but with low latency and computational overhead. SpikeSEG is the first spiking event-based segmentation network and over three experiment test achieves promising results with 96% accuracy overall and a 74% mean intersection over union for the segmentation, all within an event by event-based framework.**

***Index Terms*—Spiking Neural Network, SNN, Convolution, STDP, Segmentation**

I. Introduction

A fundamental aspect of image understanding is semantic segmentation [1], [2] , building on the progress of recognition and detection systems to not only identify but localise features in a scene. Recent research has shown convolutional neural networks (CNN) to be well suited to this task with State-of-the-art semantic segmentation deep convolutional neural networks (DCNNs) combine two separate modules: the encoder and the decoder. The encoder module uses a combination of convolution and pooling operations to extract DCNN features. The decoder module recovers the spatial details from the sub-resolution features, and predicts the object labels (i.e. the semantic segmentation) [3].

However, the underlying issues of deep learning for this approach lie within the sheer computational and time complexity. Due to run-time complexity semantic segmentation typically takes hundreds of milliseconds to run with only more recent models edging closer to real time implementations [4], [5]. However, this problem is only exacerbated in application scenarios especially on deployable or embedded systems [6]. Considering real-time semantic segmentation has important applications, e.g., street scene understanding, autonomous driving and augmented reality for wearables. Research on

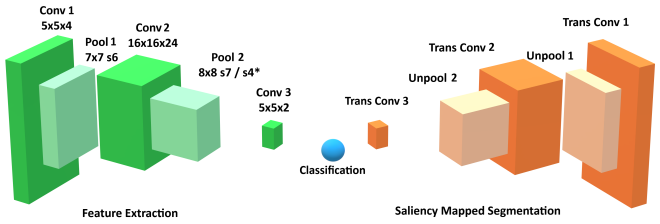


Fig. 1: SpikeSEG Network Diagram: Sizes refer to filter sizes used, with s referring to the stride, as two different strides where used in testing the alternative was marked with *

accelerating semantic segmentation is a popular area, which is seeing the use of common techniques including: network pruning, distillation, quantization, compression, factorization of standard convolution and efficient redesign of DCNNs to cut down on runtime. [3]–[5], [7] Some with specific focus on real-time application, speed and memory [4], [5]. Within our approach we suggest not just a redesign, but a systematic rethinking of task. The aim similar to the aforementioned research is to find the most efficient way to process the visual information, our approach just takes inspiration from cortical visual processing system with a Neuromorphic engineering approach [8]. A paradigm shift away from synchronous and discrete processing, Neuromorphic engineering first aims to reduce the amount of redundant visual information with a Neuromorphic event-based Sensor (EBS) [9], [10] and exploits the asynchronous spike-based output, with a similar modality processing structure of the Spiking Deep Convolutional Neural Network (SpikeCNN) [11].

In this paper we present a novel spike-based event segmentation system SpikeSEG seen in Figure 1, where the proposed framework can exploit the sparse event driven nature of a Neuromorphic Vision Sensor to deliver fast and accurate class segmented spiking images. This is due to SpikeSEG's spiking fully convolutional version of a SpikeCNNs ability to deal with the events in a timely manner. The system shows the vast reduction in computational complexity that could be delivered by an end to end spiking segmentation approach,

while maintaining the temporal advantage of an EBS and feature extraction ability of a CNN.

The rest of the paper is as follows: Section 2 covers the background on SNNs and event-based sensors, section 3 introduces several current implementations of the key component of our contribution. Section 4 gives an overview of the contribution of this paper in context, section 5 covers the experimental set-up with section 6 showing the results and section summarises with a conclusion.

II. Background

A. Spiking Neural Networks

Asynchronous spiking event-based computations like SNNs only compute on the currently active parts of the network, which in comparison to Artificial Neural Networks (ANN) can achieve orders of magnitude lesser power consumption [12]. SNNs differ from normal computation processing and take inspiration from closer to biology, where expensive memory access operations are negated due to computations and memory being exclusively local [13]. Instead of using numerical representations like traditional methods, SNNs use spikes to transmit information with a key emphasis on the timing of those spikes. A number of methods exist to train SNNs, with recent implementations seeing a conversion from CNN to SNN [14]–[17] yield promising results and open SNN architectures to the wider Machine and Deep Learning audience. However, this method is still burdened with the training computational overhead and does little to utilise the efficiency of event driven computations. The SNN's Spike Time Dependent Plasticity (STDP) and spike-based back-propagation learning have been demonstrated to capture hierarchical features in SpikeCNNs [18]–[23] Both of these methods better equip the network to deal with event driven sensors, where the significant gains over CNNs could be realised.

B. event-based Vision

As discussed, to best exploit this asynchronous event-based computation, a change to an asynchronous event-based sensing modality would also be required. Conveniently, Neuromorphic Vision Sensors (NVS) *(event-based Vision Sensors*) [10], [24] have become more popular and widespread. These camera-like devices are bio-inspired vision sensors that attempt to emulate the functioning of biological retinas. As opposed to conventional cameras, which record all the information the sensor sees at set intervals, these sensors output only when a change is detected by the sensor. So, instead of the capturing the luminosity at a set point in time, a NVS allows for a continuous temporal derivative of luminosity to be output. Whenever this happens, an event $e = [x, y, ts, p]$ is created indicating the x and y position along with the time ts at which the change has been detected and its polarity, $p \in \{1, -1\}$ i.e., if the brightness a positive or negative change. The result is a sensor able to produce a stream of asynchronous events that sparsely encodes changes with microseconds resolution and with minimum requirements in terms of power consumption and bandwidth. The growth in popularity of these type of sensors, and their advantages in terms of temporal resolution and reduced data redundancy, have led to fully exploit the advantages of event-based vision for a variety of applications albeit not using SNNs, e.g., object tracking [25], [26].

III. Related Work

A. Adaptive Neuron Thresholding

Adaptive Neuron Thresholding is a common practice within the SNN and often seen as a necessity with STDP [27]. It was also shown in [28] the importance of thresholds when converting deep neural networks to spiking, as the hierarchical layers clearly have a cascading affect. Research by Falez et al [23] showed using Time target threshold adaptation, within a STDP trained 2 Conv Layer network provided promising results. A method where an ideal time to fire is learned and used to adapt the thresholds according to try and achieve this. Most approaches use homoeostasis mechanisms based on intrinsic plasticity [29]. [27] was the only research which used synaptic scaling with recent research [17] showing a modern interpretation and combining with an intrinsic rule. Our approach takes a novel engineering approach to synaptic scaling to serve as a homoeostasis rule.

B. Neuromorphic Vision Sensor - Dynamic Vision Sensor

Neuromorphic Vision Sensors have been used successfully in multiple research areas within traditional computer vision and have utilised CNNs for Classification, Motion Estimation and Optical Flow [9]. Though some of the traditional computer vision techniques aim to exploit the sparse event driven nature of the sensor, there has been less focus on this with the CNN approach [30]–[32]. The asynchronous CNN fcYOLE [31] in particular aims to replicate benefits of a SNN by converting an already trained CNN into an asynchronous version with two methods. First leaky surface which essentially works as a leaky integrate and fire buffer layer, and second a change in the convolution and pooling layers to allow then to only process areas activated by events similar to a SNN.

C. Spiking Neural Networks

STDP has been shown to be a useful learning mechanism for unsupervised learning with a biological context with good results in [18], [23] showing its ability within the 2-3 conv layer range to be able to learn extract useful features for classification type task. STDP as the name suggests, directly accounts for time explicitly, meaning it is well suited for learning from asynchronous sensor, with single layer networks having shown impressive results [22], [33], [34]. SpikeCNNs have also shown to be able to be used a auto-encoders [11], [35] providing the only examples of a non-converted spiking auto-encoders. Panda and Roy shows some interesting results and a step forward for SpikeCNNs in terms of network complexity with multiple conv and pool layers, but has yet to be tested with an event driven input.

D. Saliency Mapping and Segmentation - Feature visualisation and mapping to pixel space

The visualisation technique in this framework uses a multi-layered Fully Convolutional Network (FCN) [3] structure which with the help of ideas from Deconvolutional Network (deconvnet) [36] and SegNet [37], to project the feature activations back to the input pixel space. More recently the deconvolution term (used to describe the backwards pass through a convolution kernel) have been referred to as Transpose Convolutions and are an essential part of most well know systems for Saliency Mapping [38] and Segmentation. The other useful development from this papers was the unpooling method of 'Switches' a way of recording the mapping in the pooling stage so that the process can be undone later.

IV. Proposed Spiking Segmentation Network - SpikeSEG

The novel process of performing semantic segmentation with a SpikeCNN and a spiking event-based image through the use of saliency maps is now presented. This method can be utilised in three ways: bounding box extraction from the segmentation extremities, pixel level segmentation and finally an intuitive understanding of how the network is working through the saliency mapping. This is to the author's knowledge the first implementation of a fully convolutional SpikeCNN network for either Segmentation or Saliency Mapping

A. System Architecture

The system can be broken down into two main sections Encoding and Decoding as seen in Figure 1, with the Encoding resembling a typical SpikeCNN but a fully convolutional version and trained with a layer wise STDP mechanism with integrate and fire neurons. The Decoding layer resemble that of similar CNN approaches for Segmentation [3], [36], [37] with unpooling and transpose convolution layers mapping the latent space class layers back into the original event pixel space. This encoding decoding structure symbolises a feature extraction then shape generation process, with a learning processing to extract common useful features, then remapping the learned features over to the shape generation process to unravel the latent space classification representation.

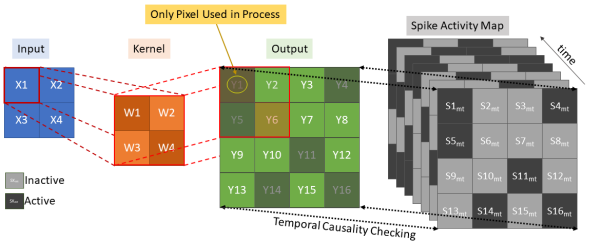


Fig. 2: Decoding using transposed convolutions with spike activity mapping, resulting in active pixel saliency mapping

B. Adaptive Neuron Threshold

A progression of the Pre-Emptive Neuron Thresholding (PENT) processes described in [39], with the adaptation now being able to affect all encoding Conv layers within the network. The thresholding is based on the homoeostasis mechanism called synaptic scaling [40], normally taking effect after hours or even days of high neuronal activity, to try and reduce activity. However, the proposed method is a pre-synaptic modifier looking at the number of events coming from the sensor seen in the buffer stage between sensor and network. Its objective is to help constrain the amount of information coming into and passing through the network. Without altering the settings of the camera, the inability to control the volume of events from the sensor needs to be managed. A constraint is set to the thresholding through experimental examination of spike propagation and misrepresentation on pretesting data, so a small variation in thresholds can be made during the training process. With pre-captured data used it is an easier constraint task with the max and min events already known for the desired buffer time. This adaptive thresholding allows the buffer to have a variable amount of events in contrast to maintaining a fixed number of events with variable buffer rate.

C. Encoding

Starting with the basics of a SpikeCNN and its STDP learning mechanism [18], a number of modifications are made to the network. These include the removal of the global pooling layer used classification, and leaving the final conv layer as the pseudo classification. The number of features maps available to the final conv layer is now mapped to the number of classes of the training data, this allows a convergence of the convolution layer to be able to delineate between shape features of the second conv layer. The input events are feed into the network via a temporal buffering stage, to allow for a more plausible current computing solution, while ideally they would just be a constant stream. To internally mimic the continuous data the buffered data is parsed into 20 steps. 10 of these are parsed event streams dividing the temporal data into equal parts, and the other 10 steps ensure the all parsed event streams have time to fully pass through network, since the network has 9 computational layers *(Conv1-Pool1-Conv2-Pool2-Conv3-TransConv3-UnPool2-TransConv2-UnPool1-TransConv1)* as seen in Figure 1. For each time step in the encoding processing a spike activity map S_{mt} is also produced, where m is the feature map and t is the time step. This allows an account of the exact spatial time location of each active pixel used in the decoding processing.

D. Decoding

The Decoding Process makes use of the same unpooling and transpose convolutions as [3], [36], [37] taking pixels in the latent classification space back into the original pixel space. No learning mechanism is required as the mapping is based on temporal active pixel saliency mapping, with the weights and switches from the encoding layer being mapped directly to

the decoding. However, a modification is required to deal with the temporal component of the spiking network, as now the latent pixel space representation must be unravelled with the constraints and context of space and time. Changes are made to both the transposed convolutions and the unpooling layers. The transposed convolution still functions as a fractionally strided convolution of the weight kernel as normal. However, now an extra step of comparing the output mapping with a temporal spike activity map of the post convolution pixel space is required as illustrated in Figure 2. This is to ascertain which pixels are allowed in line with the temporal causality gathered from the twinned encoding layer with the matching time differential of that decoding layer. Where S_{mt} is the encoding layer spike map, the decoding layer uses the equivalent t from encoding processing step, such that $t = t - t_p$, where t_p is the processing time steps between the equivalent layers. A similar process for the unpooling also takes place utilising the switch variables, however each switch now in the decoding is linked to the equivalent encoding layer through the same value of t and t_p.

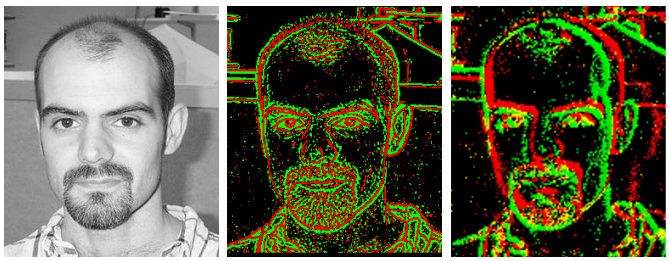
V. Experimental Set-up

The testing of the segmentation network was split into three sections to test its validity on the variety of different input data types currently used with SNNs. The three sections are split into one synthetic event dataset, and two real NVS event datasets. The first dataset, the synthetic one is a Difference of Guassian (DoG) filtered version of the CalTech Face Motorbike dataset [18] we will call DoG CalTech. The filter is used to generate pseudo events from the contrast edges of the image being converted into latencies through an inverse of their intensity.

The second dataset used actual event from a NVS, called N-CalTech [41]. This dataset is a NVS recorded version of the CalTech dataset where we extracted the same Face and Motorbike images. A NVS was used to record the images off a screen for 300ms with three saccade camera movements. A comparison of the two event images compared to the original is seen in Figure 3 with (a) the original dataset image, (b) the synthetic DoG filtered image and (c) the N-CalTech face image.

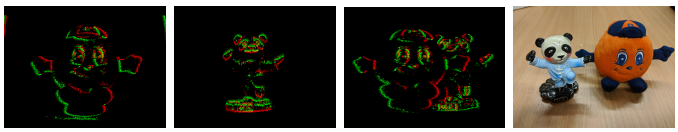
The last dataset called NVS-OrangePanda, was a self collected binary classification of two desk ornaments using a handheld NVS, the DVS346red. This dataset allowed testing of the SpikeSEG network on event inputs generated on textured 3D objects, which in comparison to the screen recorded N-CalTech provide a more realistic output of the NVS. It also allowed event capture to test the segmentation of partially overlapped classes. Examples from this dataset are shown in Figure 4, with (a) showing the University of Syracuse Orange Mascot, (b) showing a Martial Arts Panda in pose, (c) showing the Orange Mascot partially obscuring the Panda and (d) showing a picture of the two objects for context.

The parameters of the SpikeSEG network are listed in Figure 1, with the only difference between the networks for the three datasets being the stride of Pool2. During testing with real NVS events a overlapping stride length of 4 was chosen compared to the non-overlapping 7 used for the synthetic DoG CalTech. This is to account for the sparse representation that the actual NVS give out in comparison to the ideal scenario with the DoG filtered images. Considering all the network are given the same Conv1 layer weights to represent Gabor features, it was known prior to the experiments that although these are ideal filters, real NVS generated events often don't reach the minimum constraint of the adaptive threshold during the first convolution. Thus propagate less information through the network meaning the pooling can lead to an over sparsification, resulting in no classification.



(a) CalTech (b) DoG CalTech (c) N-CalTech

Fig. 3: Image of a face from CalTech Face Motorbike dataset along with the DoG filtered version (synthetic) and N-Caltech version (NVS captured)



(a) (b) (c) (d)

Fig. 4: (a-c) Images of a NVS-OrangePanda dataset, desk ornaments captures from hand-held DVS-346, 30ms integration time and (d)Picture of objects for reference

VI. Results

This sections breaks down the results from each of the three experiments, with a breakdown of overall accuracy of the network given along with mean Intersection over Union (mIoU) for the segmented bounding boxes. A illustration of a typical classification for each class is also given with a network specific latent pixel space representation, with corresponding active pixel saliency mapped segmentation output and layer specific features maps.

A. Synthetic Events - DoG CalTech

The SpikeSEG network performs well on the synthetic dataset with accuracy values of 97% and mIoU of 74%, with a the results of all experiments shown in Table I . This accuracy is inline with results from [18], [23] with a slight drop in accuracy expected when coverting to a fully convolutional network. The mIoU accuracy provides a good return for

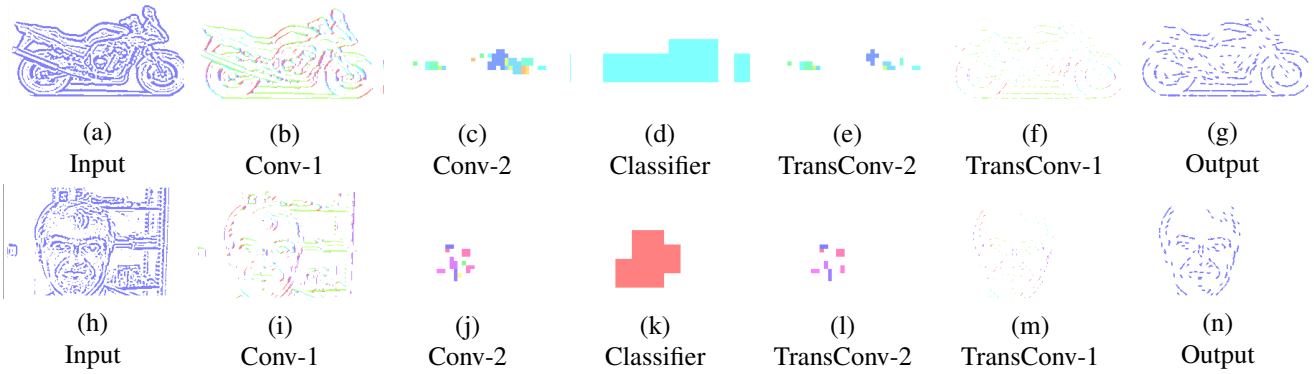
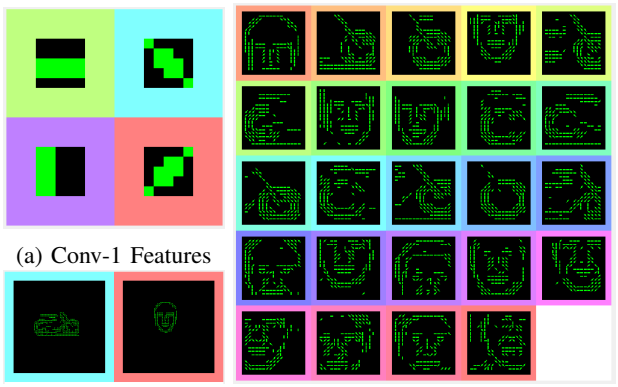


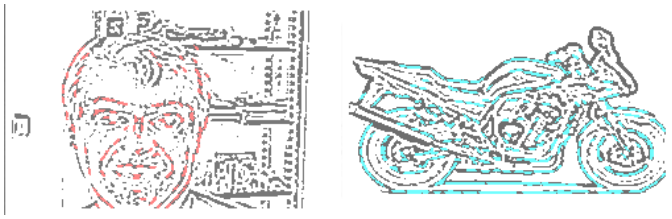
Fig. 5: Showing the latent space to segmentation mask of the DoG-CalTech face examples with pooling layers omitted for illustrative purposes. (a) and (h) shows the spiking event input to the network per image, (b-d) and (i-k) show the encoding layers of Conv-1 and 3 making a positive classifications of a face and bike, (e-f) and (l-m) show the decoding phase with Transposed Conv-2 and 1 showing the most salient features allowed with the active pixel mapping. (g) and (n) showing the most salient features mapped back to pixel space as a segmentation mask.



(a) Conv-1 Features

(b) Conv-3 Features (c) Conv-2 Features

Fig. 6: Features map representations of the convolution layers, with colouring to match the latent space representation



(a) DoG Segmentation for Face (b) DoG Segmentation for Bike

Fig. 7: Segmentation overlays for the (a) Face and (b) Motorbike class from the CalTech dataset synthetic events

segmentation extremity based bounding box estimation, with mIoUs of >50% seen as acceptable and over 75% being close to state of the art in some application. Thought considering the vast reduction in pixels due to the NVS this could be seen as an easier task.

The average amount of spiking events per image for this dataset is the highest value of all the experiments (when taking into consideration pixel size) due to the DoG filter giving a large number of edges to be converted into spiking events. As these events have no real temporal significance and are only an estimation of where the NVS would be likely to see a contrasting edge it over estimates the amount of spiking data. However, the dataset is easy to generate and allows an insight into what type of features and network parameters would work for the real NVS spiking events. As previous mentioned only the stride of the second pooling layer was altered between real and synthetic data. Both the face and motorbike latent pixel space representations with corresponding features maps are shown in Figures 5 and 6 with the saliency mapped segmentations shown in Figure 7. The latent space representation show how the features are collected in the encoding phase and how they map back to the pixel space through the decoding stage. The pooling layers are omitted in the illustration to save on space. Fig 5 (b) to (f) showing the encoding layers and (i) to (m) showing the decoding layers, with (a) and (h) showing the input events and (g) and (n) showing output segmentations. The decoding stage is a lossy upsampling due to the max pooling sparsification meaning less features are represented in the decoding phase as illustrated in Figure 5 (b-f) and (i-m). There is also a reduction in displayed features due to the temporal causality as features in section Fig 5 (b-c) and (i-j) of the image may have occurred after the corresponding classification pixel of conv-3 Fig 5 (d) and (k). The coloured pixels in each of the layers of Figure 5 are linked to the features shown in Figure 6 to allow for better understanding of the features processed in the network. The segmentation outputs of Fig 5 (g) and (n) are also seen again superimposed upon the input image in Figure 7, showing the areas where the active pixel saliency mapping segments as the given class face or motorbike. These results are inline with that the features maps in Figure 6 show to be as learned important features from the STDP training.

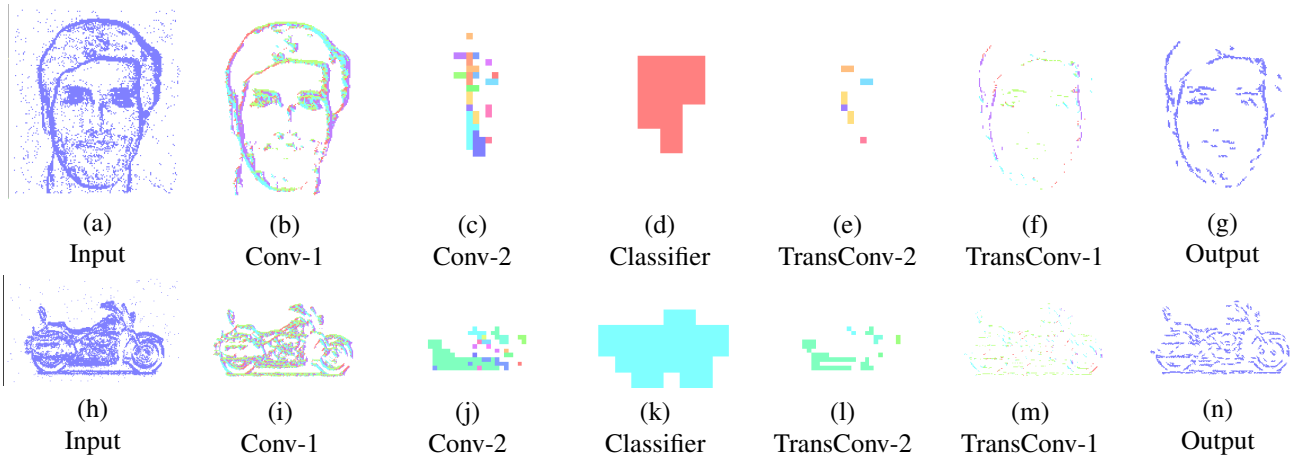
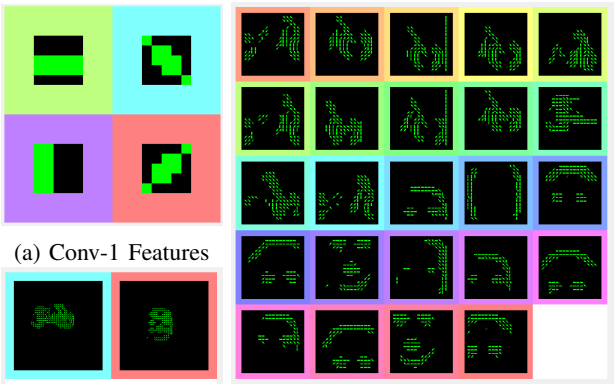


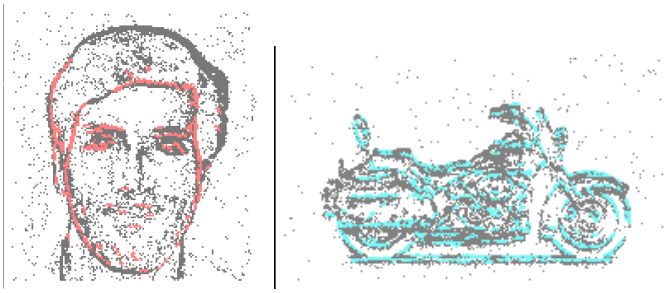
Fig. 8: Showing the latent space to segmentation mask of the N-CalTech face examples with pooling layers omitted for illustrative purposes. (a) and (h) shows the spiking event input to the network with a 30ms buffer, (b-d) and (i-k) show the encoding layers of Conv-1 and 3 making a positive classifications of a face and bike, (e-f) and (l-m) show the decoding phase with Transposed Conv-2 and 1 showing the most salient features allowed with the active pixel mapping. (g) and (n) showing the most salient features mapped back to pixel space as a segmentation mask.



(a) Conv-1 Features

(b) Conv-3 Features (c) Conv-2 Features

Fig. 9: Features map representations of the convolution layers, with colouring to match the latent space representation



(a) Segmentation for Face (b) Segmentation for Bike

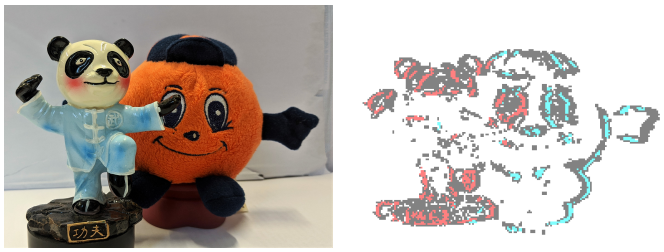
Fig. 10: Segmentation overlays for the (a) Face and (b) Motorbike class from the N-CalTech dataset events

B. NVS Screen Recording - N-CalTech

The second experiment uses a subsection of the N-Caltech dataset [41] using the same face and motorbike data as the synthetic data for easier comparison. SpikeSEG reaches accuracy values of 92% and mIoU of 67%, with all results shown in Table I .

In comparison to the synthetic data each integration phase of the event data from the NVS has on average less spiked events when testing using a buffer-size of 30ms. Although often less than 10ms of actual time was required, especially during the saccade movement of the camera, to gain a positive classification, a key benefit of the SpikeSEG is the asynchronous processing of an asynchronous input. When compared to a fixed time or fixed event number for processing, this method should always lead to the least amount of information to gain a classification, as it can operate essentially on an event by event basis. The segmentation performance along with the latent space representations and features maps are shown in Figure 9. Considering many of the parameters of the network are the same as the previous experiment the actual feature maps produced within Figure 9 for this dataset and Figure 6 for the synthetic data are reasonably different. The features from the N-CalTech data seem less detailed with face parts shown in Fig 9 (c) looking more like crude outlines in comparison to Fig 6 (c). This might help to put the performance drop of this dataset into context. Though considering the lower performance the mIoU score still reaches a performance to give merit to the SpikeSEG network with Figure 10 showing successful segmentations of both the face and motorbike. Examining the results further shows the drop, especially in mIoU often down to missing sections in the segmentation mapping, with usual culprit being the chin section for the face and back wheel for the motorbike. This insight is reflected

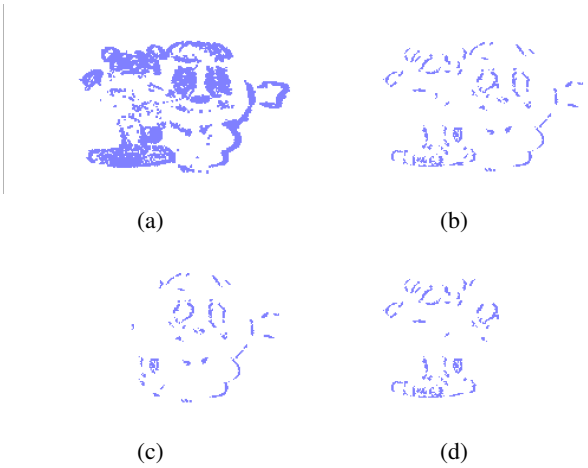
in the displayed features of Fig 9 (b) showing more eyes plus top of head for the face like features and feature that resembles the front wheel of the motorbike. Given that low latency in segmentation networks for CNNs are difficult to achieve these results show the potential a fully spiking event driven segmentation network could have when paired with a NVS.



(a) Actual image of the overlapping objects

(b) Segmentation of the two objects (red) Panda (blue) Orange Mascot

Fig. 11: Segmentation overlays for the overlapped segmentation example, with (a) showing a photo illustrating the actual scene and (b) showing the segmentation overlaid on the spiking image



(a) (b) (c) (d)

Fig. 12: (a) Original Spikes, followed by segmentations masks of (b) All Classes, (c) Orange, and (d) Panda

TABLE I: Results from the three Datasets

Dataset	Classification Accuracy (%)	mean Intersection of Union (%)
DoG CalTech	97	74
N-CalTech	92	67
NVS-OrangePanda	99	81

C. DVS346 Data - Orange/Panda

The last experimental dataset tested uses NVS-OrangePanda. This data wanted to test the validity of the network on 3D objects as the curvature and textures of the objects directly affect the light interaction, therefore will affect the perceived contrast at the edges. SpikeSEG was able to deliver accuracy results of 99% and mIoU of 81%, with all results shown in Table I .

A further test carried out within this dataset was cases of overlapping objects. SpikeSEG was able to successfully segment the scene when the classes have a spacial overlap of around 50%, with an example shown in Figure 11. Fig 11 (a) shows a colour image of the scene, while 11 (b) shows the NVS image with overlaid red and blue segmentation masks for each class. The individual class segmentations can also be seen in Figure 12. With the original spiking input shown in (a), the fully output of the system (b) and the semantic segmented masks for the two classes Orange (c) and Panda (d).

Our network was able to show that a SpikeCNN in fully convolutional format can perform segmentation using only clustered features formed from STDP. Giving this is the first implementation it provides a good argument for SpikeCNNs use for segmentation especially is low latency and low computation overhead are important factors.

VII. Conclusion

This paper presented a new spiking fully convolutional neural network used for semantic event-based image segmentation through the use of active pixel saliency mapping, SpikeSEG. Utilising an event driven neuromorphic vision sensor in an efficient manner to provide promising results on the mean intersection over union, while maintaining a temporal advantage through the event driven nature of the processing. We also show how adaptive thresholding allows for the network to deal with the wide variety of event in an effective manner, though further implementations of homoeostasis methods would only improve the performance. SpikeSEG provides a low latency and computational overhead network to perform event by event-based segmentation exploiting the feature extraction abilities of a CNN and combining with the temporal and computational benefits of a SNN. Overall the results within this paper show that SpikeCNNs can be utilised for semantic segmentation with overall accuracies of 96% and mIoU of 74%. when coupled with an sensor have the ability to deliver accuracy, low latency and a low computational overhead in one neuromorphic sensing and processing package.

References

[1] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ADE20K dataset," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua. Institute of Electrical and Electronics Engineers Inc., nov 2017, pp. 5122–5130.

[2] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8693 LNCS, no. PART 5. Springer Verlag, 2014, pp. 740–755.

[3] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2015, pp. 3431–3440. [Online]. Available: http://ieeexplore.ieee.org/document/7298965/

[4] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Bisenet: Bilateral segmentation network for real-time semantic segmentation," in *ECCV*, 2018.

[5] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid Scene Parsing Network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jul 2017, pp. 6230–6239. [Online]. Available: http://ieeexplore.ieee.org/document/8100143/

[6] M. Siam, M. Gamal, M. Abdel-Razek, S. Yogamani, M. Jagersand, and H. Zhang, "A Comparative Study of Real-Time Semantic Segmentation for Autonomous Driving," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, jun 2018, pp. 700–70 010. [Online]. Available: https://ieeexplore.ieee.org/document/8575250/

[7] R. P. K. Poudel, U. Bonde, S. Liwicki, and C. Zach, "Contextnet: Exploring context and detail for semantic segmentation in real-time," in *BMVC*, 2018.

[8] W. et.al., "Demonstrating Advantages of Neuromorphic Computation: A Pilot Study," *Frontiers in Neuroscience*, vol. 13, mar 2019. [Online]. Available: https://www.frontiersin.org/article/10.3389/fnins.2019.00260/full

[9] G. Gallego, T. Delbruck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis *et al.*, "Event-based vision: A survey," *arXiv preprint arXiv:1904.08405*, 2019.

[10] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 120 dB 15micro s Latency Asynchronous Temporal Contrast Vision Sensor," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, 2008. [Online]. Available: http://ieeexplore.ieee.org/document/4444573/

[11] P. Panda and K. Roy, "Unsupervised regenerative learning of hierarchical features in Spiking Deep Networks for object recognition," in *Proceedings of the International Joint Conference on Neural Networks*, vol. 2016-Octob. Institute of Electrical and Electronics Engineers Inc., oct 2016, pp. 299–306.

[12] J. Park, S. Ha, T. Yu, E. Neftci, and G. Cauwenberghs, "A 65k-neuron 73-Mevents/s 22-pJ/event asynchronous micro-pipelined integrate-and-fire array transceiver," in *IEEE 2014 Biomedical Circuits and Systems Conference, BioCAS 2014 - Proceedings*. Institute of Electrical and Electronics Engineers Inc., dec 2014, pp. 675–678.

[13] H. Paugam-Moisy and S. M. Bohte, "Computing with Spiking Neuron Networks," in *Handbook of Natural Computing*, G. Rozenberg, T. Back, and J. Kok, Eds. Springer-Verlag, Sep. 2012, pp. 335–376. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01587781

[14] Y. Cao, Y. Chen, and D. Khosla, "Spiking deep convolutional neural networks for energy-efficient object recognition," *International Journal of Computer Vision*, vol. 113, no. 1, pp. 54–66, 2015.

[15] E. Hunsberger and C. Eliasmith, "Spiking deep networks with lif neurons," *arXiv preprint arXiv:1510.08829*, 2015.

[16] A. Sengupta, Y. Ye, R. Wang, C. Liu, and K. Roy, "Going deeper in spiking neural networks: Vgg and residual architectures," *Frontiers in neuroscience*, vol. 13, 2019.

[17] S. Kim, S. Park, B. Na, and S. Yoon, "Spiking-yolo: Spiking neural network for real-time object detection," *arXiv preprint arXiv:1903.06530*, 2019.

[18] S. R. Kheradpisheh, M. Ganjtabesh, S. J. Thorpe, and T. Masquelier, "STDP-based spiking deep convolutional neural networks for object recognition," *Neural Networks*, vol. 99, pp. 56–67, mar 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608017302903

[19] T. Masquelier and S. R. Kheradpisheh, "Optimal Localist and Distributed Coding of Spatiotemporal Spike Patterns Through STDP and Coincidence Detection," *Frontiers in Computational Neuroscience*, vol. 12, p. 74, sep 2018. [Online]. Available: https://www.frontiersin.org/article/10.3389/fncom.2018.00074/full

[20] T. Masquelier and S. J. Thorpe, "Unsupervised learning of visual features through spike timing dependent plasticity," *PLoS computational biology*, vol. 3, no. 2, 2007.

[21] P. Panda, G. Srinivasan, and K. Roy, "Convolutional Spike Timing Dependent Plasticity based Feature Learning in Spiking Neural Networks," Tech. Rep., 2017. [Online]. Available: https://arxiv.org/pdf/1703.03854.pdf

[22] P. O'Connor, D. Neil, S.-C. Liu, T. Delbruck, and M. Pfeiffer, "Real-time classification and sensor fusion with a spiking deep belief network." *Frontiers in neuroscience*, vol. 7, p. 178, 2013. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/24115919 http://www.ncbi.nlm.nih.gov/pubmed/24115919

[23] P. Falez, P. Tirilly, I. Marius Bilasco, P. Devienne, and P. Boulet, "Multi-layered Spiking Neural Network with Target Timestamp Threshold Adaptation and STDP," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, jul 2019, pp. 1–8. [Online]. Available: https://ieeexplore.ieee.org/document/8852346/

[24] C. Brandli, R. Berner, Minhao Yang, Shih-Chii Liu, and T. Delbruck, "A 240 x 180 130 dB 3 \mui s Latency Global Shutter Spatiotemporal Vision Sensor," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 10, pp. 2333–2341, oct 2014. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6889103

[25] T. Delbruck and P. Lichtsteiner, "Fast sensory motor control based on event-based hybrid neuromorphic-procedural system," in *2007 IEEE International Symposium on Circuits and Systems*. IEEE, may 2007, pp. 845–848. [Online]. Available: http://ieeexplore.ieee.org/document/4252767/

[26] A. Glover and C. Bartolozzi, "Robust visual tracking with a freely-moving event camera," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, sep 2017, pp. 3769–3776. [Online]. Available: http://ieeexplore.ieee.org/document/8206226/

[27] K. D. Carlson, M. Richert, N. Dutt, and J. L. Krichmar, "Biologically plausible models of homeostasis and stdp: stability and learning in spiking neural networks," in *The 2013 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2013, pp. 1–8.

[28] Y. Hao, X. Huang, M. Dong, and B. Xu, "A biologically plausible supervised learning method for spiking neural networks using the symmetric STDP rule," *Neural Networks*, vol. 121, pp. 387–395, jan 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608019302680

[29] W. Zhang and D. J. Linden, "The other side of the engram: experience-driven changes in neuronal intrinsic excitability," *Nature Reviews Neuroscience*, vol. 4, no. 11, pp. 885–900, 2003.

[30] Q. Liu, G. Pan, H. Ruan, D. Xing, Q. Xu, and H. Tang, "Unsupervised AER Object Recognition Based on Multiscale Spatio-Temporal Features and Spiking Neurons," Tech. Rep.

[31] M. Cannici, M. Ciccone, A. Romanoni, and M. Matteucci, "Asynchronous convolutional networks for object detection in neuromorphic cameras," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[32] A. Linares-Barranco, A. Rios-Navarro, R. Tapiador-Morales, and T. Delbruck, "Dynamic vision sensor integration on fpga-based cnn accelerators for high-speed visual classification," *arXiv preprint arXiv:1905.07419*, 2019.

[33] O. Bichler, D. Querlioz, S. J. Thorpe, J.-P. Bourgoin, and C. Gamrat, "Extraction of temporally correlated features from dynamic vision sensors with spike-timing-dependent plasticity," *Neural Networks*, vol. 32, pp. 339–348, 2012.

[34] S. J. Thorpe, "Spike-based image processing: Can we reproduce biological vision in hardware?" in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7583 LNCS, no. PART 1. Springer Verlag, 2012, pp. 516–521.

[35] K. S. Burbank, "Mirrored STDP Implements Autoencoder Learning in a Network of Spiking Neurons," *PLoS Computational Biology*, vol. 11, no. 12, 2015.

[36] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.

[37] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, dec 2017. [Online]. Available: https://ieeexplore.ieee.org/document/7803544/

[38] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[39] P. Kirkland, G. Di Caterina, J. Soraghan, Y. Andreopoulos, and G. Matich, "Uav detection: a stdp trained deep convolutional spiking neural network retina-neuromorphic approach," in *International Conference on Artificial Neural Networks*. Springer, 2019, pp. 724–736.

[40] N. S. Desai, "Homeostatic plasticity in the cns: synaptic and intrinsic forms," *Journal of Physiology-Paris*, vol. 97, no. 4-6, pp. 391–402, 2003.

[41] G. Orchard, A. Jayawant, G. K. Cohen, and N. Thakor, "Converting static image datasets to spiking neuromorphic datasets using saccades," *Frontiers in neuroscience*, vol. 9, p. 437, 2015.