EvSegSNN : Segmentation sémantique neuromorphique pour la vision événementielle

Dalia Hareb, Jean Martinet

EvSegSNN : Segmentation sémantique neuromorphique pour la vision événementielle

D. Hareb J. Martinet

Université Côte d'Azur, CNRS, I3S, Sophia Antipolis, France

hareb@i3s.unice.fr

Résumé

La segmentation sémantique est une tâche importante de la vision par ordinateur, utile notamment pour la compréhension des scènes et la navigation des véhicules autonomes et des drones. Plusieurs variantes d'architectures de réseaux neuronaux profonds dédiées à cette tâche ont été conçues. Cependant, en raison de leurs énormes coûts de calcul et de leur forte consommation de mémoire, ces modèles ne sont pas destinés à être implémentés sur des systèmes à ressources limitées. Afin de pallier ce probléme, nous introduisons une approche de segmentation sémantique inspirée de la biologie en combinant des réseaux de neurones impulsionnels (SNN, une alternative de faible puissance aux réseaux de neurones classiques) avec des caméras événementielles dont les données de sortie peuvent directement alimenter les entrées de ces réseaux de neurones.

Nous avons conçu EvSegSNN, une architecture d'encodage-décodage en forme de U, biologiquement plausible, qui s'appuie sur des neurones Parametric Leaky Integrate and Fire afin d'optimiser l'équilibre performances/utilisation des ressources. Les expérimentations menées sur DDD17 montrent que EvSegSNN permet d'atteindre de meilleures performances que le modèle le plus proche de l'état de l'art en termes de MIoU tout en réduisant le nombre de paramètres par un facteur de 1.6 *et en épargnant la batch normalisation au modéle.*

Mots Clef

Réseaux neuronaux impulsionnels, caméras événementielles, segmentation sémantique

Abstract

Semantic segmentation is an important computer vision task, useful in particular for scene understanding and navigation of autonomous vehicles and UAVs. Several variations of deep neural network architectures have been designed to tackle this task. However, due to their huge computational costs and their high memory consumption, these models are not meant to be deployed on resource-constrained systems. To address this limitation, we introduce an end-to-end biologically inspired semantic segmentation approach by combining Spiking Neural Networks (SNNs, a low-power alternative to classical neural networks) with event cameras whose output data can directly feed these neural networks inputs. We have designed EvSegSNN, a biologically plausible encoder-decoder U-shaped architecture relying on Parametric Leaky Integrate and Fire neurons in an objective to trade-off resource usage against performance. The experiments conducted on DDD17 demonstrate that EvSegSNN outperforms the closest state-of-the-art model in terms of MIoU while reducing the number of the parameters by a factor 1.6 *and sparing a batch normalization stage.*

Keywords

Spiking neural networks, event cameras, semantic segmentation

1 Introduction

Neuromorphic computing guided by the principles of biological neural computing and inspired by the human brain's interaction with the world has been studied for years to open up the possibility of extending the use of artificial neural networks from hardware with huge computational costs and high memory consumption to embedded systems equipped with power-constrained components such as Internet of Things (IoT) devices, automotive and AR/VR. In this sense, models like Spiking Neural Networks (SNNs) are strongly representative. Because, they take an additional level of inspiration from biological neural systems compared to standard Artificial Neural Network (ANN) by integrating the concept of time into their operating model [6].

These neural models are based on timestamped discrete events called spikes. A commonly used neuron model (activation function) for SNNs is the Leaky Integrated-and-fire (LIF) neuron (Fig. 1). In a discrete formulation, the membrane potential V_i^t is measured by summing up the decayed membrane potential from previous time-steps V_i^{t-1} with a sum of the weighted spike signals W_{ij} triggered by the presynaptic neurons j :

$$V_i^t = \lambda V_i^{t-1} + \sum_j W_{ij} \cdot \theta_j^t \tag{1}$$

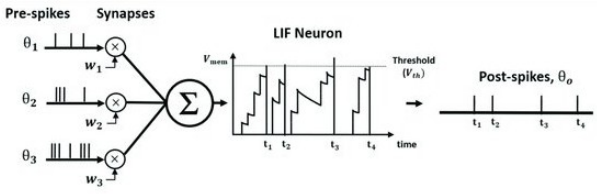


FIGURE 1 – LIF neuron (Image taken from [15].)

When this membrane potential V_i^t exceeds the firing threshold V_{th}, a spike S_i^t is generated by the neuron i :

$$\theta_i^t = \begin{cases} 1 & \text{if } V_i^t > V_{th} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

After a spike is emitted, V_i^t value is either lowered by the amount of the threshold (soft reset) or it is reset to the minimum voltage such as zero (hard reset).

Besides of taking advantage of low power consumption as they consume energy only when spikes are triggered and low computation latency, due to the asynchronous computation of the spikes and the speed of their spread [18]. SNNs are the most suitable interface of event based sensors such as event cameras.

Event cameras are bio-inspired sensors designed to overcome the standard frame cameras drawbacks. Instead of recording the video as a sequence of dense frames with every image collected with a constant rate [26], they asynchronously measure per-pixel brightness changes, and output a stream of $< x, y, p, t >$ events that encode the time t, location (x, y) and sign of the brightness changes, named polarity p (positive if the pixel brightness increases, negative otherwise).

Thus, thanks to their structure, these offer multiple advantages over standard cameras [10], such as a high temporal resolution (in the order of μs), very high dynamic range ($> 120\ dB$ vs. $60\ dB$ for frame-based cameras), a low latency ($10\ ms$ on the lab bench, and sub-millisecond in the real world) and low power consumption ($100\ mW$ max). In addition, they are extremely useful when coupled with the SNNs, because the generated events/spikes can directly feed the SNNs' inputs.

In this paper, we introduce EvSegSNN, a spiking Convolutional Neural Network (CNN) combined with event data to tackle the semantic segmentation problem. This computer vision task is mostly used in self-driving cars and UAVs that require a real-time processing with a reduced energy consumption. The proposed approach is biologically plausible, namely because it relies on SNNs, and more importantly, it does not require batch normalization [11], that is a standard process used in deep learning to fix the means and variances of a network layers by applying on a mini-batch $(x_1, x_2, ..., x_i, ..., x_m)$ of size m the following equation :

$$BN_{\gamma,\beta}(x_i) = \gamma x_i' + \beta \tag{3}$$

where γ and β indicate the parameters to be learned and x_i' is :

$$x_i' = \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}} \tag{4}$$

Here ϵ is a constant added to the mini-batch variance for numerical stability, μ and σ^2 represent the mini-batch mean and the mini-batch variance, respectively. They are measured as follows :

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x_i \tag{5}$$

$$\sigma^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu)^2 \tag{6}$$

The proposed model is validated with DDD17 dataset, an event dataset featuring driving sequences. Our main contributions are :

1. We design a spiking light Unet model for semantic segmentation that outperforms the state-of-the-art model with 5.58% of absolute MIoU while reducing the number of parameters by 62%.
2. We successfully train a large spiking neural network using Surrogate Gradient Learning.
3. We propose a biologically plausible spiking neural network Unet model that does not require the batch normalization in any layer.

The paper is organized as follows. In section 2, we introduce some research works that inspired our contribution. In section 3, we explain our methodology by describing our network architecture including the main differences with the state-of-the-art model that we take as a baseline [12]. Then, in section 4, we present the data pre-processing, the implementation details and the performance results in terms of MIoU, accuracy and number of parameters in order to compare our EvSegSNN model to the state-of-the-art model with and without batch normalization on the one hand, and to the original Unet designed for semantic segmentation on the other.

2 Related work

Semantic segmentation is a computer vision task which consists of assigning a label, corresponding to a given class, for each pixel in the image (Fig. 2). In recent years, this challenging task has been resolved using deep learning approaches based on different variations of encoder-decoder CNN architectures where the encoder downsamples the image given as input and the decoder upsamples the result returned by the encoder until it reaches the original size of the image. Among these techniques we have : Fully Convolutional Network (FCN) [16], a model obtained after transforming the fully connected layers in the most adapted classification networks (AlexNet [14], VGG16 [21] and GoogLeNet [24]) into fully convolutional ones in order to output spatial maps instead of classification scores. SegNet [2] which uses a pre-trained VGG16

without the fully connected layers as encoder and a decoder which aims to upsample the image to its original size using the max pooling indices sent by the encoder. Unet [19], a U-shaped encoder-decoder network architecture, which consists of four encoder blocks that half the spatial dimensions and double the number of filters and four decoder blocks that do the opposite process, meaning it doubles the spatial dimensions and half the number of feature channels. Both are connected via skip connections. And Xception [5] which stands for "Extreme Inception", a 36 convolutional layers split in 3 flows such that the input, first, goes through the 1st flow, then through the 2nd flow which is repeated eight times, and goes finally through the last exit flow. As input data, all these previous works have used only images captured by classic cameras. The only research work exploring the use of event based cameras to do semantic segmentation we can find, is Alonso's et al. [1]. Their model consists on an encoder-decoder architecture which has been inspired from current state-of-the-art semantic segmentation CNNs, slightly adapted to use the event data encoding. An encoder represented by Xception model in which all the training is concentrated, and a light decoder connected to the encoder via skip connections to help deep neural architecture avoid the vanishing gradient problem and to help the decoder build an image using the fine-grained details learnt in the encoder.

However, since event-driven cameras produce spikes asynchronously, their data are more suitable as input for a Spiking Neural Network which also works with spikes asynchronously. So far, this combination have been used for fundamental computer visions tasks like image classification [4], [25]. To the best of our knowledge, the only work that extends its use to semantic segmentation and we take as a reference for being the closest to our work, is Kim et al. [12]. They designed a spiking Fully Convolutional Networks that consists of an encoder-decoder architecture connected via skip connections using for each convolution layer Batch-Normalization-Through-Time (BNTT) [13] that is a batch normalization applied at every timestep. In their experiments, they use LIF neurons and piece-wise

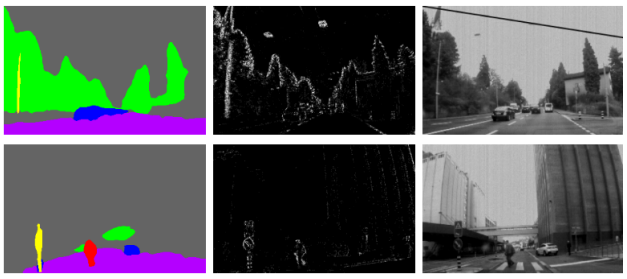


FIGURE 2 – Semantic segmentation results (left) of accumulated events within an interval of $50ms$ (middle). Gray scale images represent the scenes captured by the event camera (right). Image from [1].

function during the forward and backward propagation, respectively with a soft reset scheme. As input, they use re-scaled image resolution of 346×200 pixels to 64×64 pixel of data provided by [1], without the mean and standard deviation channels measured by computing the arithmetic mean and standard deviation of the normalized timestamps of events happening at each pixel (x_i, y_i) within an interval of $50ms$, computed separately for the positive and negative events. As result, their model reaches good performances in terms of MIoU, however the use of the batch normalization makes their spiking model non bioplausible [20].

3 EvSegSNN for semantic segmentation of event data

Our work is grounded on a Unet model [19] adapted for Spiking Neural Network (Fig. 4B). To reduce its complexity, inspired by [23] who proposed a light Unet model consisting on a smaller Unet with less parameters than the original, we design a light Unet topology EvSegSNN with four depth levels, as illustrated in Fig. 4A.

Compared to the original Unet, our light Unet model spares 4 convolution layers, 1 max-pooling layer, and 1 upsampling layer – represented within the pink dashed box of the Fig. 4B. This design choice has been made for two main reasons : (1) the deepest layer represents the heaviest part in Unet in terms of number of parameters, and (2) the deepest Unet layer in the pink dashed box is particularly suited for dense 3-channel RGB data, while event data is sparse and it features only two polarities. Hence, by lowering the number of layers in the encoder whose role is to compress the input through the layers to keep the most important details, no performance drop can be noticed on the event frames processing. In addition, its size has been reduced so as to obtain a dimension equal to 64×64 in the first layer. In order to keep the biological plausibility of SNNs, our model does not include any kind of normalization, unlike our baseline model [12] that uses Batch-Normalization-Throught-Time technique at each convolutional layer.

The computing graphs of the model is described in Fig. 3 showing that each neuron's membrane potential is measured by combining the previous membrane potential (horizontal lines) and the previous layer (vertical lines). The output spikes across all time-steps are accumulated to generate a 2-dimensional probabilistic map in the last layer whose number of channels is equal to the number of segmentation classes and whose activation function is set to *None*.

More particularly, our model is made up of Parametric LIF (PLIF) neurons introduced by [8], that are known to be less sensitive to the initial values of the learnable time constants compared to LIF. Also, it uses a soft-reset scheme because, unlike a hard reset, this scheme helps retain the residual information after the reset, thus avoiding information loss. During its training, because of the behavior of PLIF neurons making the gradient of their output spikes with respect

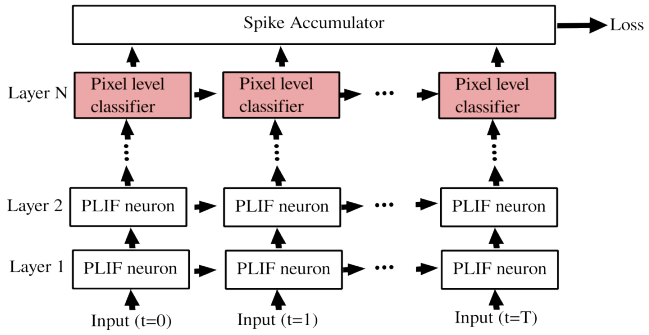


FIGURE 3 – Computational graph of both encoder decoder parts unrolled over multiple time-steps. Adapted from [12].

to their membrane potential non-differentiable, the training optimization via back-propagation of the loss is performed based on surrogate Back-propagation Through Time (BPTT) :

— *Surrogate Back-propagation* refers to the use of the backward gradient approximation using the piecewise quadratic surrogate spiking function defined in Eq. 7 and taken from [7] documentation.

$$\frac{\delta S}{\delta V} = \begin{cases} 0 & |V| > \frac{1}{\alpha} \\ -\alpha^2|V| + \alpha & V \leq \frac{1}{\alpha} \end{cases} \tag{7}$$

— *Through Time* refers to the accumulated gradients over all time-steps across all layers.

Where S, V, and α represent the output spike, the membrane potential and a scalar parameter to control smoothness of gradient, respectively.

The loss L is a pixel-wise classifier loss that is measured by summing the error between the estimated pixels' classes and the ground-truth :

$$L = -\frac{1}{N}\sum_{i=1}^{H}\sum_{j=1}^{W}\sum_{k=1}^{C} y_{i,j} log\left(\frac{e^{V_{i,j}^{k}}}{\sum_{m=1}^{C} e^{V_{i,j}^{m}}}\right) \tag{8}$$

Here, N is the size of the minibatch, C is the number of classes and H, Wrepresent the height and the width of the pseudo-frame, respectively. $y_{i,j}$ is the ground-truth label, and $V_{i,j}^{m}$ stands for the membrane potential of the neurons corresponding to the pixel (i, j) in the last layer.

4 Experimental validation

4.1 Event-based dataset

The DAVIS Driving Dataset 2017 (DDD17) [3] contains 40 different driving sequences of event data captured by an event camera. Since the original dataset provides only both grayscale images and event data without semantic segmentation labels, we used the segmentation labels provided by alonso et al. [1] who generate an approximated set of labels using a CNN model trained with Cityscapes dataset. These labels are not as perfect as if it's manually annotated and this can be clearly noticed in the left column of each of the 3 samples illustrated in Fig. 5. This dataset includes 20 different sequence intervals taken from 6 of the original DDD17 sequences. Furthermore, since only multi-channel representation of the events (normalized sum, mean and standard deviation for each polarity) are made available, we extracted the original events from DDD17 with the original $< x, y, p, t >$ structure using DDD20 tools [1] and selected events corresponding to the frames with a ground-truth.

We have used a 5-fold cross validation strategy in our experiments with 80%-20% train-test split. This choice has been made due to the fact that the original split from [12] and [1] revealed a large difference between train and test performances corresponding to a MIoU equal to 63.91% and 32.92%, respectively.

4.2 Implementation

Hyperparameters. For the experimental validation, we first implemented Kim's et al. [12] model based on the open source code [2] associated to their research work [13] related to Batch-Normalization-Throught-Time. The hyperparameters of the model are available in their paper [12].

We implemented our model using Spikingjelly [7], a Python framework based on Pytorch dedicated to SNNs. We trained the model on a desktop computer equipped with an NVIDIA RTX A5000 GPU card using the hyperparameters given in Table 1.

TABLE 1 – EvSegSNN hyperparameters

Optimizer	Nadam
Learning rate	$2e-3$
Learning rate scheduling	$[8, 16, 24, 50]$ with a factor of 10
Batch size	16
Number of epochs	70
Timesteps	20
Leak factor	0.99
Membrane threshold	1.0

Performance metrics. To evaluate our model and compare it to the baseline, we use standard metrics of semantic segmentation as used by Alonso et al. [1] i.e. *accuracy* and *Mean Intersection over Union (MIoU)* :

$$\begin{aligned} accuracy(y, \widehat{y}) &= \frac{1}{N}\sum_{i=1}^{N}\delta(y_i, \widehat{y}_i) \\ &= \frac{TP + TN}{TP + TN + FP + FN} \end{aligned} \tag{9}$$

1. DDD tools : https ://github.com/SensorsINI/ddd20-utils.
2. https ://github.com/Intelligent-Computing-Lab-Yale/BNTT-Batch-Normalization-Through-Time.

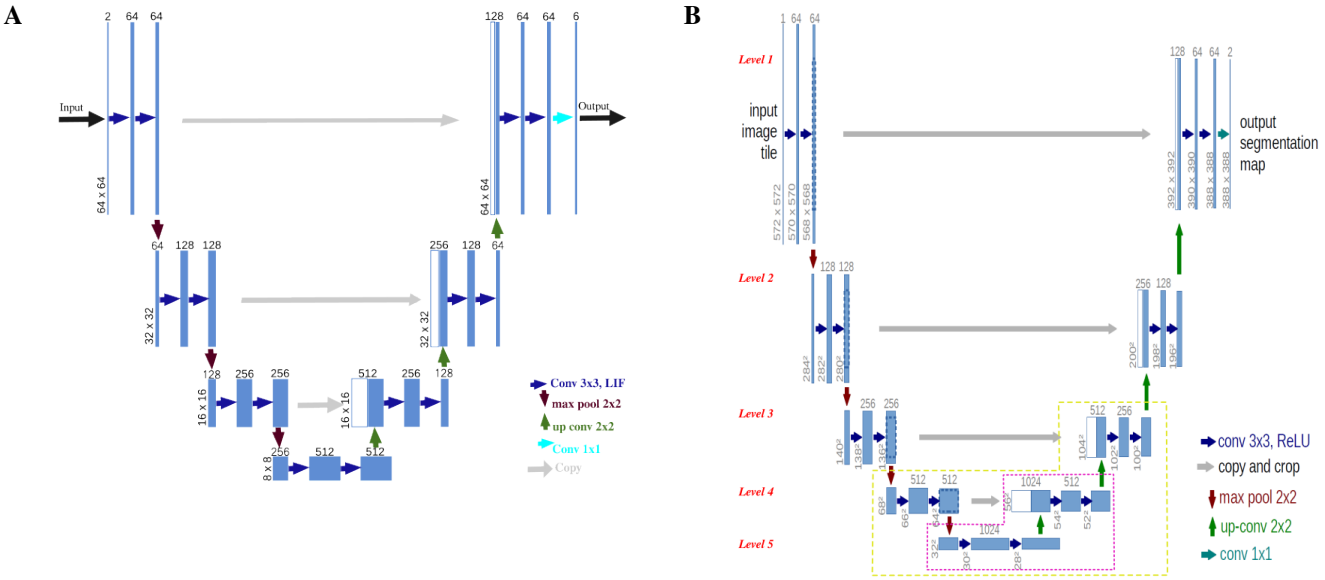


FIGURE 4 – A : The proposed EvSegSNN obtained after reducing the Unet size and its depth by one level corresponding to the layers within the pink box. B : The original Unet model [19]. The pink and yellow boxes corresponds to 2 depth levels reduced to obtain 2 light Unet models.

$$MIoU(y,\widehat{y}) = \frac{1}{C}\sum_{j=1}^{C}\frac{\sum_{i=1}^{N}\delta(y_{i,c},1)\delta(y_{i,c},\widehat{y_{i,c}})}{\sum_{i=1}^{N}max(1,\delta(y_{i,c},1)\delta(\widehat{y_{i,c}},1))} = \frac{TP}{TP+TN+FP+FN} \tag{10}$$

In Eq. 9 and Eq. 10, y and $\widehat{y}$ are the desired output and the system output respectively, C is the number of classes, N is the number of pixels and δ denotes the Kronecker delta function, TP, TN, FP and FN stand for : true positive, true negative, false positive and false negative, respectively.

Moreover, we compare our model to the baseline in terms of *number of parameters* whose reduction involves a reduction of the runtime as well as the energy consumption, which represents the first motivation of using SNNs.

4.3 Performance results

Figure 6 shows a performance comparison of the proposed EvSegSNN (blue) and the baseline model with and without Batch-Normalization-Through-Time (green and red, respectively) in terms of MIoU and accuracy, with respect to the number of parameters. It is interesting to note that, in the one hand, despite the fact that the number of parameters of our model ($8.55M$) is reduced by a factor of 1.6 compared to Kim et al. [12], the performances of both models remain very close. On the other hand, considering the bio-plausibility of Spiking Neural Networks, we notice that EvSegSNN built without BNTT yields a MIoU of 45.54% and an accuracy of 89.90%, while the baseline designed without BNTT only obtains a MIoU of 39.96% and an accuracy of 86.03%. Therefore, in addition to be bio-plausible, EvSegSNN outperforms the baseline model without BNTT while having a number of parameters 1.6 times smaller. Note that the number of parameters of Kim et al. [12] model without BNTT is slightly reduced compared to the original one (with BNTT) due to the removal of trainable parameters used for BNTT.

By taking a close look at MIoU results of EvSegSNN during training and testing summarized in Table 2, we can see that the MIoU of the *Background*, the *Road*, the *Tree* and the *Car* classes show good performances, while the *Panel* and the *Pedestrian* classes show unsatisfying values. This can be explained by the average number of pixels belonging to each class in all the DDD17 frames. Indeed, in Table 3, we can see that the four first classes are represented by a large amount of pixels while the *Panel* and *Pedestrian* classes contain in average less than 500 and 100 pixels respectively, thus explaining the poor performances for both these classes.

TABLE 2 – Class-wise MIoU results for EvSegSNN.

	Road	Background	Panels	Tree	Pedestrian	Car
MIoU Train	83.93	93.01	**17.26**	75.92	**14.81**	72.75
MIoU Test	79.14	89.26	**6.25**	48.54	**2.44**	47.63

TABLE 3 – Average number of pixels per class in DDD17

	Road	Background	Panels	Tree	Pedestrian	Car
Pixels	7.49K	47.22K	**0.476K**	10.55K	**0.087K**	3.36K

Figure 7 presents a comparison of the proposed model with the original Unet in terms of number of parameters and performances. The original Unet is taken as a reference model for EvSegSNN with the maximal set of parameters, thus scoring 100% in each metric. Note that, similarly to EvSegSNN, the input size of the original Unet has also been

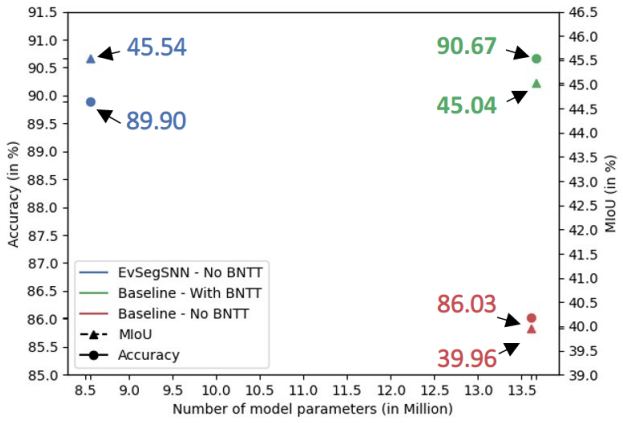
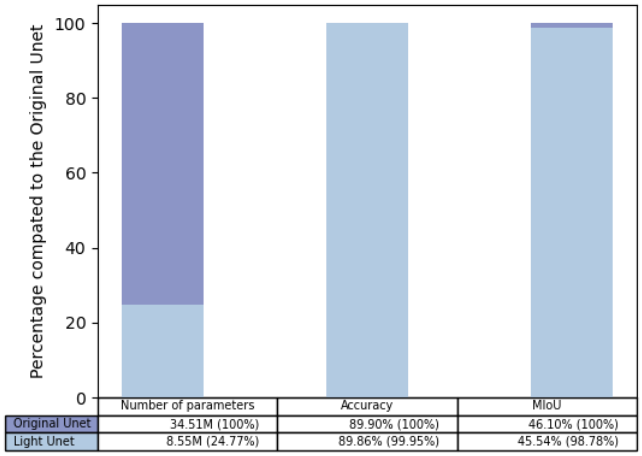


FIGURE 6 – MIoU and accuracy of EvSegSNN and the baseline (Kim et al. [12]) with and without BNTT w.r.t. the number of parameters.

reduced to fit the 64×64 size of the dataset.

As we can see, by designing EvSegSNN whose number of trainable parameters is equal to $8.55M$ (about 25% the number of parameters of the original Unet), we obtain performances in terms of MIoU and accuracy almost equal to the original. This result shows that we successfully designed a smaller and less complex Unet while maintaining the performances. Thus, this also shows that our initial hypothesis stating that due to its structure, event data does not require a deep encoder to preserve the most important details while being down-sampled is verified.



	Number of parameters	Accuracy	MIoU
Original Unet	34.51M (100%)	89.90% (100%)	46.10% (100%)
Light Unet	8.55M (24.77%)	89.86% (99.95%)	45.54% (98.78%)

FIGURE 7 – Comparison of light Unet (EvSegSNN) to the Original

Note that if we go further in the simplification and introduce another light Unet with much less parameters ($2M$) by sparing the layers represented within the yellow dashed box of the Fig. 4B, we notice a significant decrease of the semantic segmentation quality whose evaluation metrics return a MIoU of 40.19% and an accuracy of 84.67% repre-

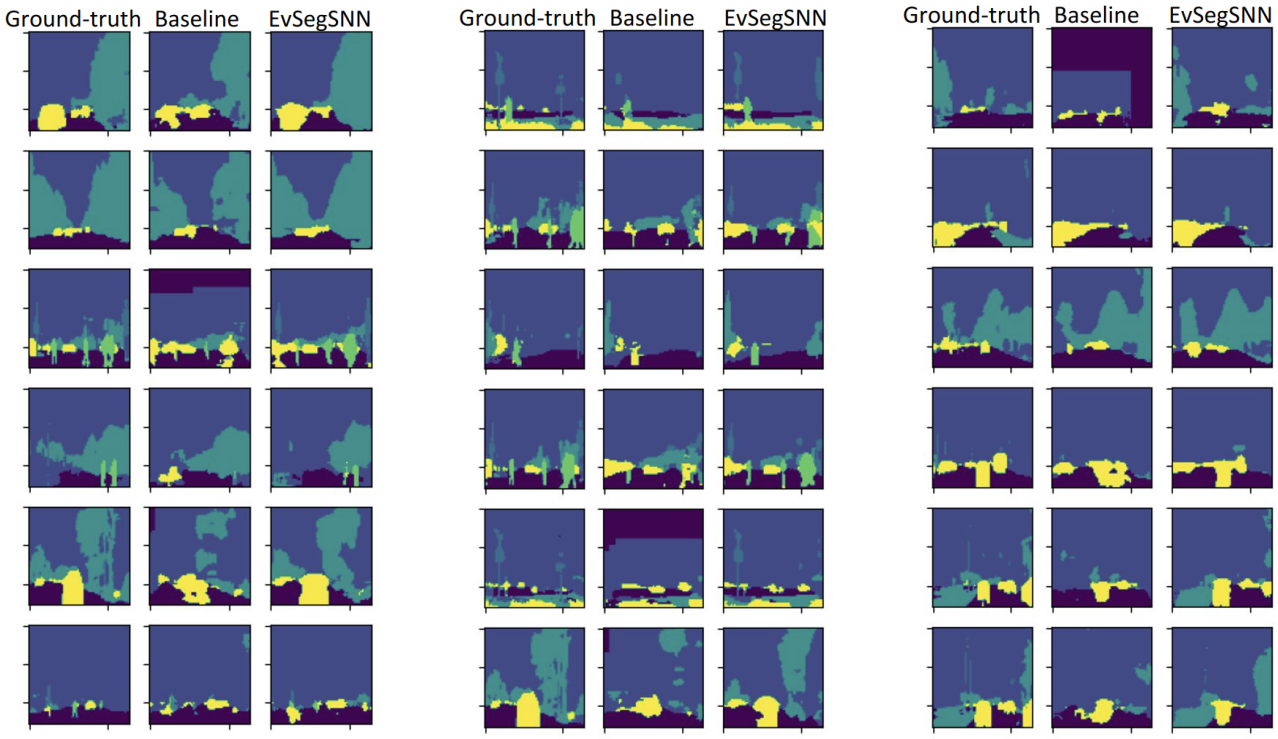


FIGURE 5 – Example of semantic segmentation results using Light Unet on the DDD17. We visualize the Ground-Truth - Baseline prediction - EvSegSNN prediction

senting a drop of 5.91% and 5.23% respectively, compared to the original Unet,.

5 Conclusion

In this paper, we introduced a bio-inspired model to tackle the semantic segmentation task that consists of a spiking light Unet architecture directly fed by an event camera. This architecture has been built using : Surrogate Gradient Learning, Parametric LIF and SpikingJelly framework. Our semantic segmentation experiments show that our approach outperforms a state-of-the-art model in terms of MIoU and number of parameters while considering the biological plausibility of Spiking Neural Network. It also shows a large variation regarding the segmentation result corresponding to each class which is due to the unbalanced number of pixels belonging to each class. Hence, for future work, it would be interesting to train and test our model using better semantic segmentation labels obtained and generated by e.g. [22]. Besides, the next step is to implement EvSegSNN on a low-power neuromorphic hardware such as Loihi [17] or SpiNNaker [9] which will enable power-efficient embedded semantic segmentation applications.

Acknowledgements

This work was supported by the European Union's ERA-NET CHIST-ERA 2018 research and innovation programme under grant agreement ANR-19-CHR3-0008.

Références

[1] Inigo Alonso and Ana C Murillo. Ev-segnet : Semantic segmentation for event-based cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet : A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12) :2481–2495, 2017.

[3] Jonathan Binas, Daniel Neil, Shih-Chii Liu, and Tobi Delbruck. Ddd17 : End-to-end davis driving dataset. *arXiv preprint arXiv :1711.01458*, 2017.

[4] Enea Ceolini, Charlotte Frenkel, Sumit Bam Shrestha, Gemma Taverni, Lyes Khacef, Melika Payvand, and Elisa Donati. Hand-gesture recognition based on emg and event-based camera sensor fusion : A benchmark in neuromorphic computing. *Frontiers in Neuroscience*, 14 :637, 2020.

[5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[6] Loïc Cordone, Benoît Miramond, and Philippe Thierion. Object detection with spiking neural networks on automotive event data. *arXiv preprint arXiv :2205.04339*, 2022.

[7] Wei Fang, Yanqi Chen, Jianhao Ding, Ding Chen, Zhaofei Yu, Huihui Zhou, Timothée Masquelier, Yonghong Tian, and other contributors. Spikingjelly. `https://github.com/fangwei123456/spikingjelly`, 2020. Accessed : YYYY-MM-DD.

[8] Wei Fang, Zhaofei Yu, Yanqi Chen, Timothée Masquelier, Tiejun Huang, and Yonghong Tian. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2661–2671, 2021.

[9] Steve Furber and Petruț Bogdan. *Spinnaker-a spiking neural network architecture*. Now publishers, 2020.

[10] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. Event-based vision : A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(1) :154–180, 2020.

[11] Sergey Ioffe and Christian Szegedy. Batch normalization : Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.

[12] Youngeun Kim, Joshua Chough, and Priyadarshini Panda. Beyond classification : directly training spiking neural networks for semantic segmentation. *arXiv preprint arXiv :2110.07742*, 2021.

[13] Youngeun Kim and Priyadarshini Panda. Revisiting batch normalization for training low-latency deep spiking neural networks from scratch. *Frontiers in neuroscience*, page 1638, 2020.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6) :84–90, 2017.

[15] Chankyu Lee, Syed Shakib Sarwar, Priyadarshini Panda, Gopalakrishnan Srinivasan, and Kaushik Roy. Enabling spike-based backpropagation for training deep neural network architectures. *Frontiers in neuroscience*, page 119, 2020.

[16] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[17] Garrick Orchard, E Paxon Frady, Daniel Ben Dayan Rubin, Sophia Sanborn, Sumit Bam Shrestha, Friedrich T Sommer, and Mike Davies. Efficient neuromorphic signal processing with loihi 2. In *2021 IEEE Workshop on Signal Processing Systems (SiPS)*, pages 254–259. IEEE, 2021.

[18] Michael Pfeiffer and Thomas Pfeil. Deep learning with spiking neurons : Opportunities and challenges. *Frontiers in neuroscience*, 12 :774, 2018.

[19] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net : Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[20] Nolan Peter Shaw, Tyler Jackson, and Jeff Orchard. Biological batch normalisation : How intrinsic plasticity improves learning in deep neural networks. *Plos one*, 15(9) :e0238454, 2020.

[21] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv :1409.1556*, 2014.

[22] Zhaoning Sun, Nico Messikommer, Daniel Gehrig, and Davide Scaramuzza. Ess : Learning event-based semantic segmentation from still images. In *Computer Vision–ECCV 2022 : 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIV*, pages 341–357. Springer, 2022.

[23] Aleazer Viannie Sunn, Aiden Langba, Hamebansan Mawlong, Alexy Bhowmick, and Shyamanta M Hazarika. Light u-net : Network architecture for outdoor scene semantic segmentation. In *Proceedings of International Conference on Frontiers in Computing and Systems : COMSYS 2021*, pages 83–92. Springer, 2022.

[24] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[25] Tasbolat Taunyazov, Weicong Sng, Hian Hian See, Brian Lim, Jethro Kuan, Abdul Fatir Ansari, Benjamin CK Tee, and Harold Soh. Event-driven visual-tactile sensing and learning for robots. *arXiv preprint arXiv :2009.07083*, 2020.

[26] Alberto Viale, Alberto Marchisio, Maurizio Martina, Guido Masera, and Muhammad Shafique. Carsnn : An efficient spiking neural network for event-based autonomous cars on the loihi neuromorphic research processor. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–10. IEEE, 2021.