HALSIE: Hybrid Approach to Learning Segmentation by Simultaneously Exploiting Image and Event Modalities

Shristi Das Biswas, Adarsh Kosta, Chamika Liyanagedera, Marco Apolinario and Kaushik Roy
Purdue University, West Lafayette, Indiana, USA
{sdasbisw, akosta, cliyanag, mapolina, kaushik}@purdue.edu

The supplementary material is organized as follows: Section **S-1** (**S**- refers to the sections in this suppl. document) investigates the denoising characteristic of SNNs; Section **S-2** discusses edge-feature extraction using ANNs vs SNNs; Section **S-3** highlights qualitative evaluations on the newly released DSEC-Semantic dataset; and Section **S-4** provides details regarding computing approximate inference cost for different methods based on the number of floating point operations they perform per cycle. Section **S-5** elaborates on the decoupled sampling rates used in our architecture. Section **S-6** finally provides some additional visualisations to inspect phenomenon we discuss in the main paper.

1. Denoising with SNNs

In order to segment objects efficiently, we aim to isolate relevant events corresponding to objects of interest in the scene, filtering out any spurious event inputs generated due to background clutter and sensor noise. Spiking neurons such as the Leaky-Integrate and Fire are excellent candidates as they are capable of maintaining an internalized state called membrane potential u_{mem}, which decays over time at a rate controlled by the leak factor. The leak factor denotes how much of the membrane potential is retained for the next time step, i.e., the higher the leak factor, the slower the rate of decay. If the accumulated membrane potential of the neuron exceeds the threshold at any point, $(u_{mem} > v_{th})$, the neuron emits an output spike and resets its membrane potential.

Spurious events due to sensor noise are usually generated at much lower rates than events triggered by moving vehicles, humans or other relevant objects of interest in a scene. As such, if the time between input events is large, the membrane potential decays its value before it can reach the threshold. However, if these input events occur more frequently, they are able to overcome the decay and increase the membrane potential towards the threshold. Thus, the neuron generates output spikes if the input events occur at a frequency higher than a certain value. We visualise this phenomenon in Fig. 1. Leveraging this sparse spiking property of LIF neurons, our SNN-based TFE module enjoys the benefits of implicit denoising of input events with no extra parametric or learning overheads in contrast to dense ANNs or RNNs which do not directly lend themselves to filtering out sensor noise.

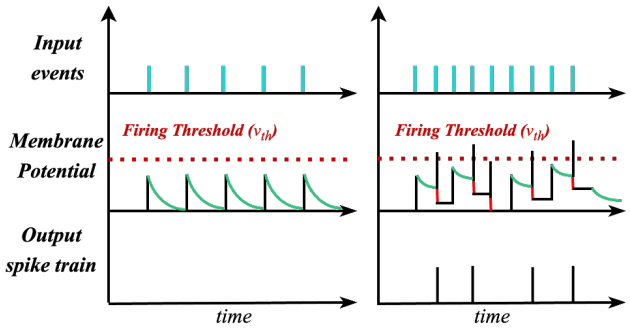

Figure 1. **Temporal sensitivity of a LIF neuron.** Left graph shows the neuron response to inputs occurring at a lower rate; right graph shows the neuron response to inputs occurring at a higher rate. Once the inputs occur at a rate higher than a certain frequency, the neuron generates spikes and resets to a resting value.

2. Sharp feature map extraction with ANNs vs SNNs

We know that in stable lighting conditions, events are triggered by moving edges (e.g., object contour and texture boundaries), making an event-based camera a natural edge extractor. We try to use this captured edge information and investigate how sparse processing of events in our SNN-based TFE is better suited to not only high energy savings, but also implicitly supports sharp edge extraction. We investigate feature maps from processing events using dense methods like ANNs instead of sparse methods like SNNs. Visualising results in Fig. 3, we find that using SNNs for extracting cues from the input results in edge-discriminative feature maps (row 2) caused by high spike generation at the object-boundary pixels where events are triggered at higher rates. In contrast, using ANN-like dense processing networks (row 3) do not offer similar benefits as is seen

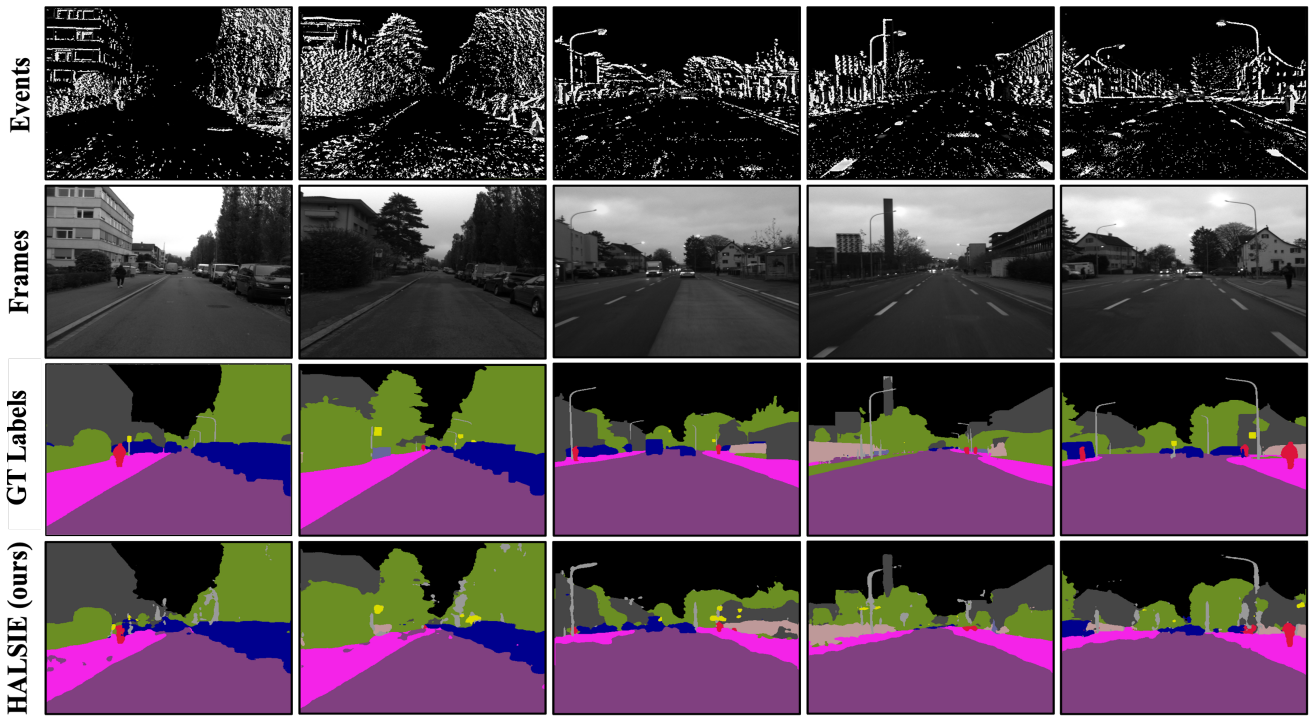

Figure 2. **Semantic segmentation results on the DSEC-Semantic dataset.** Best viewed in color.

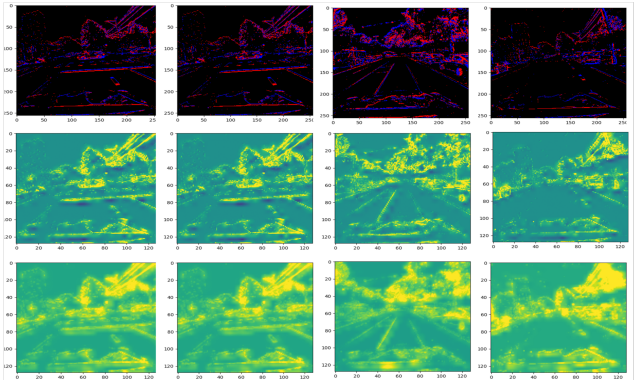

Figure 3. **Feature maps** generated by processing Top row: events inputs with Middle row: SNN-based TFE and Bottom row: ANN-based SFE.

in Fig. 3 with the feature maps looking fuzzier.

3. Evaluation on DSEC-Semantic Dataset

We provide qualitative samples of HALSIE on the DSEC-Semantic dataset [1]. Fig. 2 shows our model can successfully detect objects in various scenes on the test set comprising 'zurich_city_13_a', 'zurich_city_14_c' and 'zurich_city_15_a' sequences. Interestingly, even with a complex dataset such as DSEC-Semantic with 11 semantic classes: background, building, fence, person, pole, road, sidewalk, vegetation, car, wall, and traffic sign, our lightweight and inference-efficient method does not fall short and is still able to segment large objects with almost accurate boundaries, and detect smaller objects with fine-grained details in a variety of driving scenes.

4. Computing Approximate Inference Cost

This section validates the energy-efficiency of our method in terms of compute intensity per inference cycle. Spiking Neural Networks (SNNs), often referred to as the third generation of neural networks, are well known for being highly energy-efficient compared to traditional Artificial Neural Networks (ANNs). For estimating the inference cost per cycle for different architectures, we try to highlight how computations in SNNs and ANNs primarily differ from each other. SNNs offer highly sparse, asynchronous event-driven ACcumulate (AC) operations over time. Hence, the synaptic computes are performed only when an input spike arrives. In contrast, ANNs perform expensive Multiply-and-ACcumulate (MAC) operations for computing dense Matrix-Vector Multiplication (MVM) functions, irrespective of the sparsity of inputs. We use the findings in [2] to specify that a MAC operation requires a total of $E_{MAC} = 4.6pJ$ of energy while an AC operation requires only $E_{AC} = 0.9pJ$ for a 32-bit floating-point computation in 45nm CMOS technology. This makes an AC operation $5.1\times$ more energy-efficient than a corresponding MAC operation. Note that comparisons on a different technology node would also generate similar energy requirement

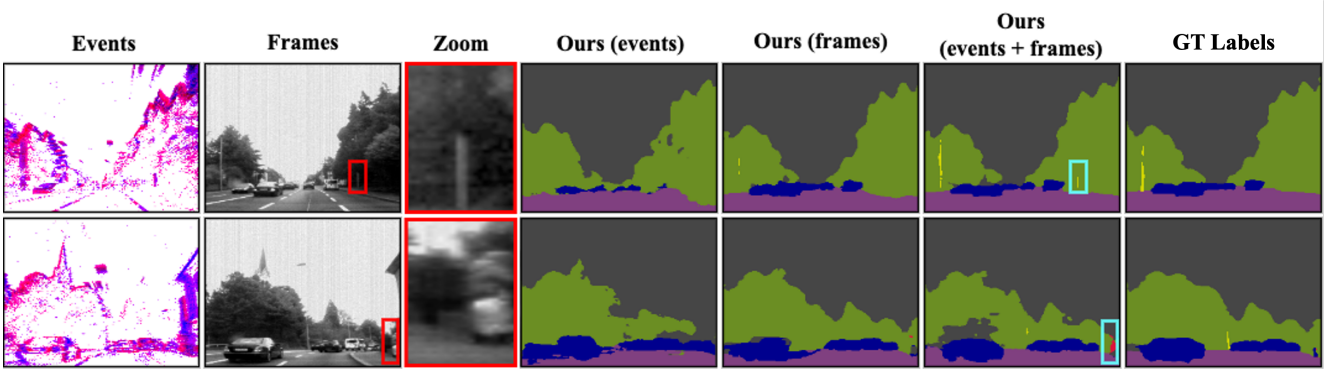

Figure 4. **Segmentation results for multi-modal vs. uni-modal settings.** Finer objects missing from the GT-labels (zoomed-in patch in the red box) are detected by our hybrid multi-modal method trained using events + frames.

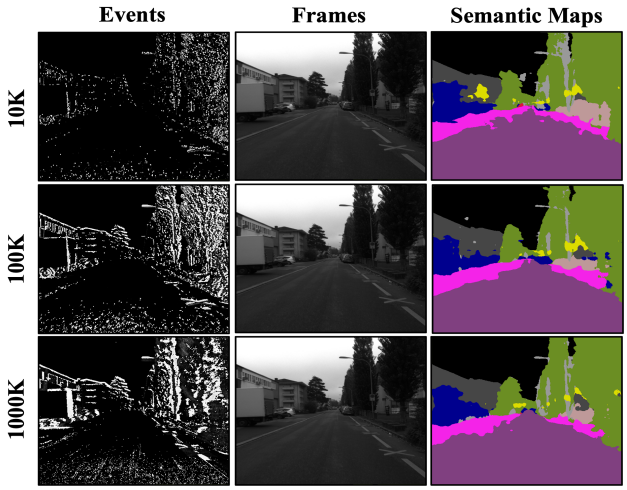

Figure 5. **Impact of event window density on predictions.** Top: $10K$; Middle: $100K$; Bottom: $1000K$. Moderate event bin density $100K$ offers better performance compared to its counterparts.

trends between SNNs and ANNs. Coupled with the number of floating point operations (FLOPs) performed by the network for a single inference, we benchmark compute energy cost of existing approaches following the estimation method used in [3–7] on a 45nm CMOS process node.

It is worth mentioning that we neglect energy consumed by the memory or any peripheral circuitry and only consider the compute cost for MAC/ AC operations. First, we calculate the total number of synaptic operations performed in each layer. For SNNs, the number of FLOPs at a layer is obtained by multiplying the mean spiking rate at each timestep for that layer, the number of synaptic connections and the number of operating timesteps. The small input spiking activities obtained in different SNN layers are mainly because of the fact that event camera outputs are highly sparse in nature and the spiking neurons generate progressively sparser outputs as the network depth increases. This sparse firing rate is essential for exploiting efficient event-based computations in the SNN layers. In contrast, ANNs execute dense matrix-vector multiplication operations without considering the sparsity of inputs. In other words, ANNs simply feed-forward the inputs at once, with a fixed total number of operations. This leads to high energy requirements (compared to SNNs) since operations are executed for both zero and non-zero input values, leading to unnecessary compute [3].

Given a number of neurons M, a number of synaptic connections C, and a mean firing activity F, the number of FLOPs at each timestep for a layer l is calculated as $M_l \times C_l \times F_l$. In the case of ANNs, we have a $mean\_spiking\_rate = 1$ and $number\_of\_timesteps = 1$ for each layer. Hence, the total compute energy cost per inference cycle can be formalized as follows:

$$FLOPs_{ANN} = \sum_l M_l \times C_l$$

$$FLOPs_{SNN} = N \sum_l M_l \times C_l \times F_l$$

$$E_{Total} = FLOPs_{ANN} \times E_{MAC} + FLOPs_{SNN} \times E_{AC}$$

where N is the number of timesteps and E_{Total} denotes the total compute cost for a single inference. We utilize the above-described formulation to estimate the total compute energy required by different networks during inference. Note that we only consider convolution operations to compute the number of FLOPs, and neglect energy consumed by Batch-norm layers, or activations after each convolution layer. The results in Tab. 1, Tab. 2 and Tab. 3 in the main manuscript suggest that our method achieves competitive performance and the least compute energy requirement compared to the 'heavier' current state-of-the-art approaches [8–12]. Specifically, we reduce parameter count ($\sim 33\times$ lower) and inference cost ($\sim 20\times$ lower) compared to prior art. This is mainly attributed to our efficient hybrid SNN-ANN temporal-spatial feature learning method which enables us to use a lightweight architecture

for edge-compute without compromising on semantic performance. Additionally, the SNN-encoder pathway in our network contributes negligibly to the total compute energy cost, reducing our energy requirements even further.

5. Architecture Details

We provide details on the decoupled sampling rates $r_h \times r_w$ of the dilated convolution blocks used in our MMix module as follows: 1×6, $(6 \times 21, 18 \times 15, 1 \times 1)$ and 6×3. The rates correspond to dilated conv. blocks appearing from left to right in Fig. 4 from the main paper. For blocks appearing in the same vertical alignment, rates are stated in top to bottom order inside single brackets.

6. Additional visualisations

Following the discussion in the main paper, we provide visualisations for segmentation results using only one modality (events or frames) versus multi-modal settings in Fig. 4 and show the impact of varying event window density on predictions in Fig. 5.

References

[1] Z. Sun, N. Messikommer, D. Gehrig, and D. Scaramuzza, "Ess: Learning event-based semantic segmentation from still images," *arXiv preprint arXiv:2203.10016*, 2022. 2

[2] M. Horowitz, "1.1 computing's energy problem (and what we can do about it)," in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pp. 10–14, 2014. 2

[3] C. Lee, A. K. Kosta, A. Z. Zhu, K. Chaney, K. Daniilidis, and K. Roy, "Spike-flownet: event-based optical flow estimation with energy-efficient hybrid neural networks," in *European Conference on Computer Vision*, pp. 366–382, Springer, 2020. 3

[4] Y. Kim, J. Chough, and P. Panda, "Beyond classification: directly training spiking neural networks for semantic segmentation," *Neuromorphic Computing and Engineering*, 2022. 3

[5] C. Lee, A. K. Kosta, and K. Roy, "Fusion-flownet: Energy-efficient optical flow estimation using sensor fusion and deep fused spiking-analog network architectures," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 6504–6510, IEEE, 2022. 3

[6] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura, *et al.*, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, 2014. 3

[7] B. Rueckauer, I.-A. Lungu, Y. Hu, M. Pfeiffer, and S.-C. Liu, "Conversion of continuous-valued deep networks to efficient event-driven networks for image classification," *Frontiers in neuroscience*, vol. 11, p. 682, 2017. 3

[8] L. Wang, Y. Chae, and K.-J. Yoon, "Dual transfer learning for event-based end-task prediction via pluggable event to image translation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2135–2145, 2021. 3

[9] I. Alonso and A. C. Murillo, "Ev-segnet: Semantic segmentation for event-based cameras," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 0–0, 2019. 3

[10] D. Gehrig, M. Gehrig, J. Hidalgo-Carrió, and D. Scaramuzza, "Video to events: Recycling video datasets for event cameras," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3586–3595, 2020. 3

[11] L. Wang, Y. Chae, S.-H. Yoon, T.-K. Kim, and K.-J. Yoon, "Evdistill: Asynchronous events to end-task learning via bidirectional reconstruction-guided cross-modal knowledge distillation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 608–619, 2021. 3

[12] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza, "High speed and high dynamic range video with an event camera," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 6, pp. 1964–1980, 2019. 3